



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

WEBOVÝ NÁSTROJ PRO SLEDOVÁNÍ POSTUPU NA ZÁLEŽITOSTECH

A WEB TOOL FOR TRACKING PROGRESS ON ISSUES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. JAN LAMACZ

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Ing. ADAM HEROUT, Ph.D.

BRNO 2018

Abstrakt

Tato práce se zabývá návrhem a tvorbou webové aplikace na řízení projektů. Vytvořená aplikace je, na rozdíl od konkurenčních produktů, orientovaná na proces vývoje projektu a zakládá si na vizuální reprezentací stavu jednotlivých úkolů.

Jednou ze základních vlastností aplikace je její jednoduchost a uživatelská přívětivost, kterou cílí mimo jiné i na uživatele, kteří by rádi využívali projektového řízení, ale nejsou ochotni učit se ovládat i jinak sebasteně složitou aplikaci.

Aplikace je vyvíjena ve frameworku Symfony.

Abstract

This diploma thesis focuses on design of the web application for project management. The final product is more process-oriented compared to most of other widely spread project management tools. It builds on the visual representation of tasks' progress.

One of the main features is its simplicity and user friendliness as it also aims at the users who are not willing to invest their time to discover more complex applications but are willing to use some project management tool.

The application is developed in the Symfony framework.

Klíčová slova

management projektů, design zaměřený na uživatele, uživatelská zkušenost, webová aplikace, Symfony, orientace na proces

Keywords

project management, user-centered design, user experience, web application, Symfony, process-oriented

Citace

LAMACZ, Jan. *Webový nástroj pro sledování postupu na záležitostech*. Brno, 2018. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce prof. Ing. Adam Herout, Ph.D.

Webový nástroj pro sledování postupu na záležitostech

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana prof. Ing. Adama Herouta, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Jan Lamacz
23. května 2018

Obsah

1	Úvod	2
2	Aplikace zaměřené na uživatele	3
2.1	User-centered design	3
2.2	Principy uživatelského návrhu	4
2.3	Proces vytváření uživatelského návrhu	5
2.4	Prvky použitelné na webových stránkách	6
3	Project Management	9
3.1	Úvod do project managementu	9
3.2	Existující řešení	10
4	Návrh aplikace	16
4.1	Návrh systému	16
4.2	Návrh uživatelského rozhraní	19
4.3	Druhá fáze návrhu	25
4.4	Technologie použitelné při vývoji webové aplikace	27
5	Implementace	29
5.1	Serverová část	29
5.2	Grafické uživatelské rozhraní	37
6	Testování	49
6.1	Způsob testování	49
6.2	Výsledky testování	50
7	Závěr	54
	Literatura	55
A	Obsah přiloženého paměťového média	57
B	Odkazy na aplikaci	58
C	Další prvky uživatelského rozhraní	59

Kapitola 1

Úvod

Tato diplomová práce popisuje způsob vzniku webové aplikace určené k řízení projektů a monitorování stavu uživatelských úkolů. Hlavním cílem vytvářené aplikace je uživatelská přívětivost a jednoduchost jejího používání.

V kapitole číslo 2 se zaměřuji na tvorbu produktů kladoucích hlavní důraz na uživatelskou přívětivost a dále v ní aplikuji obecnou teorii na prostředí webových aplikací zaměřených na uživatele. V další kapitole číslo 3 se poté zabývám teorií řízení projektů a zkoumám tři zástupce již vytvořených webových aplikací v této oblasti, jimiž jsou Asana, Podio a Trello. U těchto aplikací hodnotím jejich pozitiva a negativa a zároveň provádím jejich porovnání.

Kapitola číslo 4 popisuje návrh vytvářené aplikace, popis cílových skupin aplikace a návrh jejích jednotlivých prvků. Tato kapitola také obsahuje popis dodatečného návrhu prvků, které byly do aplikace přidány až na základě výstupu testování.

Samotná implementace navrženého systému je poté popsána v kapitole číslo 5. V kapitole číslo 6 popisují způsob testování vytvořené aplikace a výstupy, které z tohoto testování vznikly.

Kapitola 2

Aplikace zaměřené na uživatele

V této kapitole je nejprve vysvětlen pojem návrhu zaměřeného na uživatele a jeho stručná historie. Dále je v sekci číslo 2.2 diskutováno šest základních principů tohoto návrhu. Sekce číslo 2.3 poté popisuje proces vytváření správného návrhu.

V sekci číslo 2.4 jsou popsány jednotlivé principy použitelné v kontextu tvorby webových stránek se silnou uživatelskou orientací.

2.1 User-centered design

User-centered design (UCD), neboli česky „design zaměřený na uživatele“, je přístup k návrhu a vytváření produktů. Cílem UCD je dosáhnout jednoduššího použití těchto produktů uživatelem, proto se při vývoji využívají principy ergonomie¹ a techniky z oblasti použitelnosti². [13]

Pojem User-centered design byl poprvé použit doktorem Donaldem Normanem již v roce 1986 v knize *User-Centered System Design: New Perspectives on Human-Computer Interaction*. [18] V této knize autor propojuje svůj inženýrský přístup s poznatky z oboru psychologie. Na jednoduchých příkladech z každodenního života vysvětluje dle jeho názoru *dobrý* a *špatný* návrh těchto předmětů.

Norman vyzdvihuje nutnost zaměřit se při návrhu těchto objektů na jejich koncové uživatele od začátku a přemýšlet o způsobech, jak budou tyto předměty používány v praxi. Ve svých publikacích mimo jiné radí zjednodušit vytvářené objekty takovým způsobem, aby jejich používání bylo v jakékoliv situaci jednoduché a intuitivní. Dále také vyzdvihuje nutnost zpětné vazby – uživatel tak může pomoci snáze odhalit případné chyby a může tím přispět k jejich nápravě. Při vývoji jakéhokoliv produktu zároveň doporučuje zamyslet se nad těmito třemi otázkami: jak se bude navrhovaný produkt používat, kdo jej bude používat a proč. [17]

V polovině 90. let 20. století byl tento princip ještě více zobecněn a vznikla metodologie nazvaná Human-centered design (HCD), neboli „design zaměřený na lidi“. [13] Zaměření metodologie se rozšířilo z pohledu na interakci uživatele se systémem na obecnější pohled, který určuje, jakým způsobem jsou lidské schopnosti a charakteristiky ovlivněny systémem mimo přímou interakci s tímto systémem či jeho rozhraním. [20]

¹Věda zabývající interakcí mezi člověkem a ostatními prvky systému a hledáním optimálního řešení problémů z pohledu potřeb člověka a pracovního prostředí. [11]

²Míra, kterou lze produkt využívat danou skupinou uživatelů za účelem dosažení předem daných cílů s ohledem na účinnost, efektivitu a uspokojení definovaných potřeb uživatele. [12]

Obě tyto metody zdůrazňují přímou účast uživatele na vývojovém procesu daného projektu a jeho vliv na hodnocení navržených řešení. S širším rozvojem webových stránek byly zároveň tyto principy přizpůsobeny různým specifikům, které s nimi souvisejí.

2.2 Principy uživatelského návrhu

Mezinárodní organizace pro normalizaci v roce 2010 vydala normu ISO 9241-210 popisující požadavky a doporučení týkající se principů navrhování interaktivních počítačových systémů zaměřených na uživatele. [13] Součástí této normy je také následujících šest klíčových principů, které by měl každý uživatelsky zaměřený systém dodržovat.

2.2.1 Návrh je založen na porozumění uživatelům a jejich záměrům

Základem tohoto principu je pochopení kontextu, v jakém bude uživatel daný produkt používat. Podle Dr. Davida Travise by se tento kontext dal rozdělit na tři části: vcítění se do jednotlivých uživatelů, pochopení důvodu, kvůli kterému daný produkt používají, a celkové okolí produktu, ve kterém je užíván. [22]

Norma zároveň uvádí příklad dvou uživatelských rozhraní: jednoho zaměřeného na mladší uživatele stahující hudbu skrze svůj telefon a druhého zaměřeného na obchodníka získávajícího obchodní informace z tabletu. Při návrhu každého z těchto dvou rozhraní by se měl klást důraz na jiné aspekty. Je tedy patrné, že řešení vyhovující jedné skupině uživatelů může být pro skupinu druhou zcela nepřijatelné.

2.2.2 Zapojení uživatelů do návrhu i vývoje produktu

Tento princip se snaží zajistit, že vývojové týmy budou spolupracovat a myslet na uživatele během celého vývojového cyklu produktu, ne například pouze na začátku návrhu, nebo naopak provedením několika dotazníků na konci vývoje produktu.

Dle normy je také nutné, aby bylo zapojení uživatelů *aktivní*, tedy aby se nejednalo pouze o schválení vytvořeného produktu na konci jednotlivých fází, ale aby byl uživatel přímo součástí samotného vývoje. [22]

2.2.3 Návrh je vylepšován uživatelsky zaměřeným hodnocením

Při vývoji produktu by se nemělo zapomínat na systematické testování uživateli. To lze provádět například poměrně nenáročným *uživatelským testováním* (anglicky „usability testing“), *testováním grafického designu*, nebo například *eyetrackingem*. [24]

Norma zároveň upozorňuje na častou chybu, kdy toto testování proběhne během vývoje pouze jednou, a to obvykle na konci vývojového cyklu. Z tohoto důvodu norma doporučuje, aby se uživatelské testy prolínaly napříč celým vývojem produktu. Je také možné testovat i pouhé koncepty, jako jsou papírové prototypy nebo elektronické náčrtky. [22]

2.2.4 Vývoj probíhá iterativně

Tento princip je v normě popsán poměrně jednoznačně: „Vhodný návrh interaktivního systému nemůže být typicky dosažen bez použití více iterací.“ [13] Tímto chtějí autoři zdůraznit, že je pro uživatele téměř nemožné najednou popsat vše, co od systému očekávají. Vývojář tedy musí uživateli ukázat svůj prvotní návrh, který sice typicky nebude splňovat všechny jeho požadavky, ale postupným iterativním zlepšováním původního návrhu půjde dosáhnout dobrého výsledku.

Například při použití vodopádového modelu³ vývoje, ve kterém je nutné ještě před fází implementace sepsat všechny požadavky na systém, půjde uživatelsky zaměřeného výsledku dosáhnout jen stěží. Ačkoliv norma nemluví o žádné konkrétní metodice vývoje, jako vhodná bývá často doporučována agilní metodika. [22]

2.2.5 Návrh musí zahrnovat celou uživatelskou zkušenost

Vytvoření uživatelsky příjemného projektu pouze začíná tím, že se jej pokusíme udělat jednoduchý k užívání (nebo se alespoň pokusíme neudělat jej složitý), zahrnuje také mnohem širší spektrum činností.

Tento princip byl do normy oproti její předchozí verzi z roku 1999 přidán nově a sama norma o něm tvrdí, že koncept použitelnosti popsany v ISO 9241 je širší a může zahrnovat druh vnímání a emocionálních aspektů, které jsou obvykle spojeny s uživatelskou zkušeností.

2.2.6 Členové týmu by měli mít mezioborové znalosti

Tímto principem chtějí autoři normy naznačit, že dobrý uživatelsky zaměřený návrh může jen těžce vzejít z týmu, který je složen jen z velmi podobně smýšlejících vývojářů. Ideální tým by tedy měl zahrnovat širokou škálu různě zaměřených členů s různým pohledem na danou tematiku od koncových uživatelů, přes marketingové experty, analytiku, až po programátory.

2.3 Proces vytváření uživatelského návrhu

Proces vytváření uživatelsky zaměřeného produktu se podle normy ISO 9241 obecně skládá z následujících čtyř aktivit, které formují hlavní cyklus práce:

1. **Specifikování kontextu používání:** V této části by se měla stanovit skupina cílových uživatelů, mělo by se určit z jakého důvodu a za jakých podmínek by chtěli službu používat.
2. **Specifikování požadavků:** Po určení kontextu je nutné stanovit konkrétní požadavky a uživatelské cíle na finální produkt. Při jejich navrhování je nutné zaměřit se mimo jiné na tržní úspěšnost finálního produktu.
3. **Vytvoření designového řešení:** Po specifikaci základních cílů a požadavků pak následuje fáze vytváření samotného řešení, která je obvykle realizována postupně od hrubého konceptu až po finální návrh.

³Sekvenční vývojový proces, ve kterém postupně, za sebou, následují jeho jednotlivé fáze: analýza požadavků, návrh, implementace, testování a údržba.

4. **Vyhodnocení návrhu:** Poslední a velmi důležitou fází, je získání zpětné vazby na nově vzniklý produkt, která se obvykle provádí například pomocí *uživatelského testování* s reálnými uživateli.

Před dokončením vývojového procesu a realizací výsledného produktu mohou být výše popsané kroky několikrát zopakovány. Tento proces může být začleněn buď do vodopádového modelu, nebo do agilní či jiné metodiky. Většinou vždy záleží na vytvářeném produktu, požadavcích, vývojovém týmu a jiných okolnostech vývoje produktu.[13] [23]

2.4 Prvky použitelné na webových stránkách

Principy návrhu zaměřeného na uživatele se dají samozřejmě použít i při vývoji webových stránek. Snadná použitelnost a užitečnost jsou při vývoji webových stránek velmi důležité faktory. Správně navržené uživatelsky zaměřené webové stránky by se měly snažit udržet uživatele *spokojeným*, měly by se snažit vypadat *dobře* a měly by mít jednoduchou navigaci. Jejich cílem by mělo být zajištění, že uživatel stránky ihned neopustí a prohlédne si, co stránka vůbec nabízí. [21]

Následující podkapitola popisuje zásady, kterými by se měly dobře navržené stránky řídit.

2.4.1 Dostupnost

Dostupný web je rychlý a nejsou na něm na první pohled vidět žádné chyby. A pokud už návštěvník na chybu narazí, je vysvětlena srozumitelným jazykem a rychle opravena. [24]

Rychlost je na webu zásadní faktor – neznamená to, že web nemůže obsahovat například velké fotografie, ale musí být na ně připraven (např. postupným načítáním stránky). Návštěvník očekává, že web bude okamžitě reagovat na jeho akce, protože není ochoten čekat. [24] Studie společnosti Akamai z roku 2009 například tvrdí, že až 40 % návštěvníků opustí webovou stránku, pokud její načítání trvá déle než tři sekundy. [6]

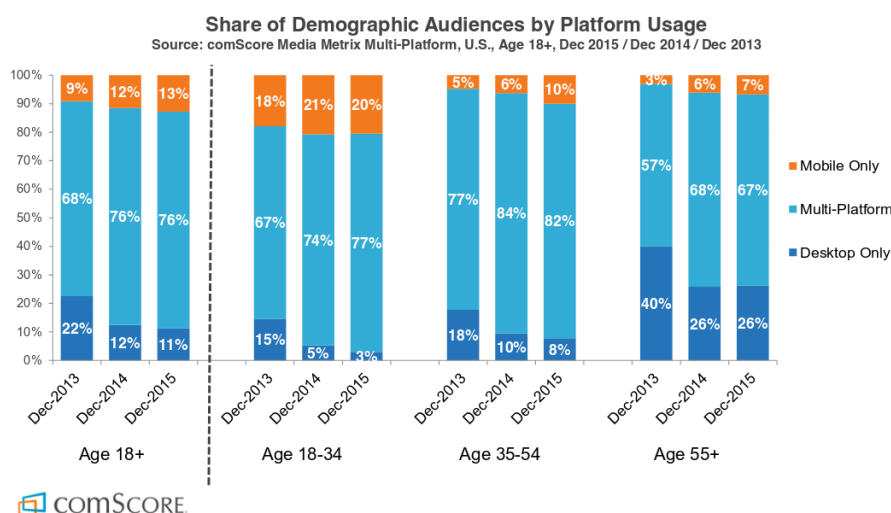
Tento přístup navazuje na tvrzení Donalda Normana z knihy *The Design of Everyday Things* [17], ve kterém tvrdí, že zpětná vazba zásadně pomáhá uživatelům vyhnout se zklamání a zmatkům. Na uživatelsky zaměřených webových stránkách by každá uživatelská akce měla vyprodukovat reakci ze strany webu. Například kliknutí na tlačítko by mělo alespoň část obsahu buďto změnit, nebo aspoň uživatele informovat o právě probíhající akci. [21]

2.4.2 Přístupnost

Smyslem přístupnosti je zmírňovat či odstraňovat bariéry na webu tak, aby kladl svým uživatelům co nejméně překážek, a ti jej mohli bez problémů používat. [24]

Podle studie společnosti ComScore z let 2013 – 2015 používá až 89 % uživatelů k zobrazení webových stránek kombinaci mobilních zařízení nebo počítače, z čehož celých 13 % používá pouze mobily, viz obrázek 2.1. Z těchto údajů jednoznačně vyplývá, že je nutné webové prezentace přizpůsobit také mobilním zařízením, která mívají obvykle menší velikost obrazovky a jiný způsob prohlížení, než klasické desktopové počítače. Zároveň je ovšem nutné si uvědomit, že je stále nutné navrhovat stránky i pro tyto stolní počítače, na kterých uživatelé, podle stejné studie, stále utrácejí větší část peněz utracených na internetu. [9]

Přístupný web by měl myslet také na lidi se zdravotním postižením. Měl by se snažit usnadnit své používání nevidomým lidem (například umožněním používání asistivních



Obrázek 2.1: Graf přístupu na webové stránky dle používané platformy [9]

technologií), vytvořením vhodných klávesových zkratk nebo dostatečným barevným kontrastem. Tyto úpravy přístupnosti však nepomáhají pouze skupině uživatelů se zdravotním handicapem, zvyšují přístupnost webu všem. [24]

2.4.3 Použitelnost

Web musí být pro uživatele lehce použitelný. Návštěvník se v něm rychle zorientuje a je schopen provést rychle to, proč na web přišel - ví, kde je, kam může pokračovat a na co lze kliknout. [24]

Důležitou součástí použitelného webu je jednoduchá a logická navigace. Bez možnosti jednoduše se pohybovat mezi jednotlivými stránkami k obsahu, který uživatel hledá, je vysoká pravděpodobnost, že uživatel naše stránky opustí. Přímá a intuitivní navigace navíc uživatele podporuje v hlubším prozkoumávání dalších oblastí naší webové prezentace. [21]

Uživatel by zároveň neměl být nucen pamatovat si, co který prvek navigace znamená – rozmístění prvků by mělo být konzistentní a nemělo by se na jednotlivých stránkách lišit. Vývojář by zároveň měl myslet na to, že uživatel může na stránky přijít odkudkoliv, a vždy by měl vědět, kde se zrovna na webu nachází. [8] V tomto bodu může také uživatelům pomoci například existence mapy stránek.

2.4.4 Důvěryhodnost

Důvěryhodný web je autentický, prověřený a garantuje výsledky. Důvěryhodnost je předpoklad pro úspěšnou transakci – na nedůvěryhodných webech uživatel jednoduše žádné konverzní akce neuskuteční. [24]

Podle studie firmy *Huff Industrial Marketing* z roku 2015 až 52 % lidí na webu očekává stránku „O nás“ a kolem 64 % návštěvníků by chtělo vidět kontaktní informace na hlavní stránce. [7] Z webu by tedy mělo být na první pohled jasné, kdo za ním stojí.

Je také vhodné předem očekávat možné dotazy návštěvníků, které mohou mít při čtení článků nebo zvažování provedení nákupu. Jedním ze způsobů, jak na tyto otázky odpovědět,

je například vytvoření stránky s Často kladenými otázkami (FAQ), které mohou pomoci překonat strach z nákupu nebo ujistit váhající uživatele. [21]

Kvalitní grafický design dokáže výrazně zvýšit důvěryhodnost webové prezentace. Jeho součástí je zejména kombinace jednoduchosti a zajímavosti: konzervativní použití barev a nebo zvýraznění nejdůležitějších prvků stránky. [21]

Kapitola 3

Project Management

Plánování a řízení projektů je lidstvu známo od nepaměti. Vždy bylo potřeba něco naplánovat, provést a zhodnotit. V posledním století se však přistupuje k racionalizaci těchto procesů a jejich zpřesnění a důkladné analýze. Tímto přístupem je možné najít slabé články v těchto procesech a celé je tak optimalizovat.

První sekce této kapitoly se okrajově zabývá teorií řízení projektu a fázemi, které jsou důležité k pochopení procesu a tedy k tomu, aby bylo možno aplikaci vyvinout s jasným cílem a požadavky. Druhá sekce kapitoly se soustředí na webové aplikace Asana, Podio a Trello, které již byly v této oblasti vyvinuty a jsou často používány.

3.1 Úvod do project managementu

Dr. Kerzner ve své knize *Project Management: A Systems Approach to Planning, Scheduling, and Controlling* uvádí srozumitelné podmínky pro definici projektu a project managementu. [15] Tyto definice pak slouží k porozumění problematice a jsou nezbytné k vypracování vlastní aplikace.

Za projekt považuje Dr. Kerzner aktivity, které mají specifický cíl, který má být splněn v rámci daných specifikací, mají pevně definovaný začátek a konec, mají limity ve financování, využívají lidské i jiné zdroje, jako jsou peníze, lidé či vybavení a jsou multifunkční. [15]

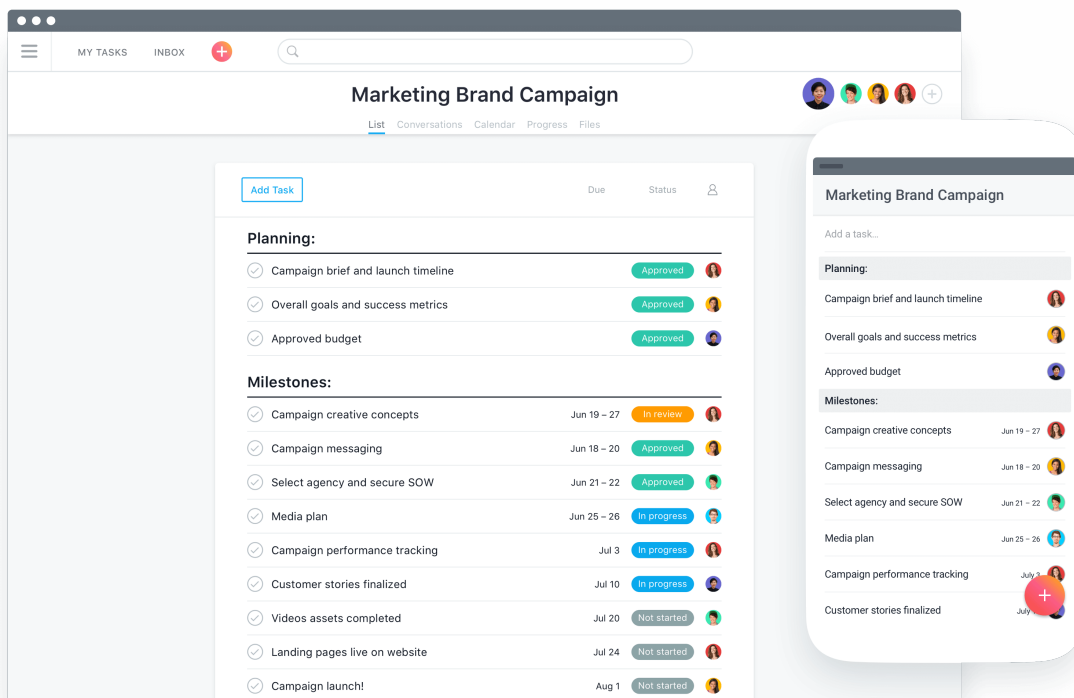
Project management, neboli řízení projektu, pak podle Dr. Kerznera zahrnuje pět různých stádií. Prvním stádiem je zahájení projektu, kdy je vybrán nejlepší projekt v závislosti na dostupných prostředcích. V této fázi je nezbytné vytyčit výhody projektu, připravit si finanční dokumenty a určit vedoucího projektu. Druhá fáze – plánování projektu – zahrnuje zpracování pracovních požadavků, definici kvality a kvantity práce, definici zdrojů, naplánování daných aktivit a zhodnocení případných rizik. Třetí fáze, která je označována za výkonnou, zahrnuje jmenování členů týmu, řízení práce a spolupráci se členy týmu na zlepšování projektu. Čtvrtou fází je monitorování a kontrola projektu, kdy je nezbytné sledovat postup, porovnávat reálný výstup s očekávaným, analyzovat rozdílnosti a jejich vliv a případně optimalizovat proces. Poslední fází je uzavření projektu, kdy dochází k ověření dosažených výsledků, finančním uzavěrkám, administrativním uzavření projektu a smluv. [15] Často autoři uvádějí jako další část projektu také zpětné ohlédnutí a hodnocení celého procesu tak, aby mohly být chyby v procesu použity jako příležitost k poučení. [10]

3.2 Existující řešení

V tuto chvíli existuje na trhu nepřehledné množství aplikací pro project management. Každá z aplikací má svá specifika, v mnohém jsou si však podobné. Některé aplikace jsou velice jednoduché a intuitivní, některé jsou naopak složitější, nicméně zajišťují možnost kompletního project/task managementu, aniž by bylo nutné použít jiné aplikace. Dá se tedy skrze ně komunikovat s členy týmu, zadávat úkoly, nahrávat soubory, plánovat schůzky a další. Z těch nejčastěji používaných jsem pro analýzu vybral tři – Asanu, Podio a Trello. U každé aplikace jsem se zaměřil na všeobecnou funkcionalitu, na její silné a slabé stránky, na identifikaci vhodné cílové skupiny a na to, které části bych pak mohl implementovat ve vlastní aplikaci, případně čeho se vyvarovat. Na konci této podkapitoly uvádím srovnání jednotlivých řešení a jejich možné využití pro jednotlivé cílové skupiny, které jsou pro mou aplikaci relevantní (viz podkapitola 4.1.3).

3.2.1 Asana

Asana nabízí pohled na projekt jak detailně, tak také z větší perspektivy. Pomocí kalendáře je také možno zobrazit úkoly i v čase. Stejně jako ostatní aplikace s podobným cílem umožňuje zadávání úkolů dalším členům týmu, přidávat soubory, komentáře, sledovat postup v úkolu a další. Aplikace je také propojena s dalšími aplikacemi, jako je Google Drive, Calendar a další. Úkoly v Asaně mohou být organizovány buďto jako seznam a nebo jako nástěnka (stejně jako Trello). Na obrázku číslo 3.1 lze vidět příklad zobrazení jednotlivých úkolů v této aplikaci.



Obrázek 3.1: Seznam úkolů týmu uživatelů v aplikaci Asana, včetně zobrazení v mobilní aplikaci. [16]

Výhodou Asany je možnost zadávat prioritu přiděleným úkolům a také sledovat statistiky zátěže jednotlivých členů týmu pomocí jednoduchých grafů, které zohledňují počet zadaných a počet splněných úkolů. Tato funkcionalita umožňuje sledovat vývoj daného projektu, bohužel však pouze v obrysech - není jasné vidět, které části projektu zaostávají či vyžadují více pozornosti. Jednotlivé úkoly se dají třídit na základě barevného systému. Výhodou je také možnost přidělit jednotlivý podúkol do různých úkolů najednou, což umožňuje obejít předpoklad, že se úkol může nacházet pouze ve stavu „splněno“ a „nesplněno“. Pomocí této funkce je možno vytvořit si Kanban nástěnku¹ (podobnou aplikaci Trello) vnořenou do Asany, kdy se jednotlivé úkoly manuálně přesunují mezi sloupce s progresem. Asana také nabízí možnost duplikace jak projektu, tak jeho podúkolů. V placené verzi lze také založit šablony projektů. V rámci motivačního aspektu aplikace nabízí občasné zobrazování jednorozce probíhajícího přes obrazovku při dokončení projektu, což může na jednu stranu působit mile, na druhou stranu však také poněkud zvláštně.

Nevýhodou je pak podpora pouze patnácti členů týmu bezplatně, což se však dá obejít formací více týmu po patnácti členech. Zároveň jsou v bezplatné verzi jen základní funkce, které je možno za poplatek rozšířit. Největší nevýhodou je však z mého pohledu fakt, že je aplikace orientována spíše na výsledek, než-li na proces. Pro jednoduché úkoly je systém ano/ne zcela dostačující, pro komplikovanější úkoly však může být nevhodný. Dalším důvodem nespokojenosti s Asanou je systém připomínek deadline, kdy posílá Asana e-mailovou připomínku týden před finálním datem, den před ním, v den ukončení projektu a dále pak každý den po překročení data, a to po dobu jednoho týdne, což je dle mého názoru zbytečné.[16]

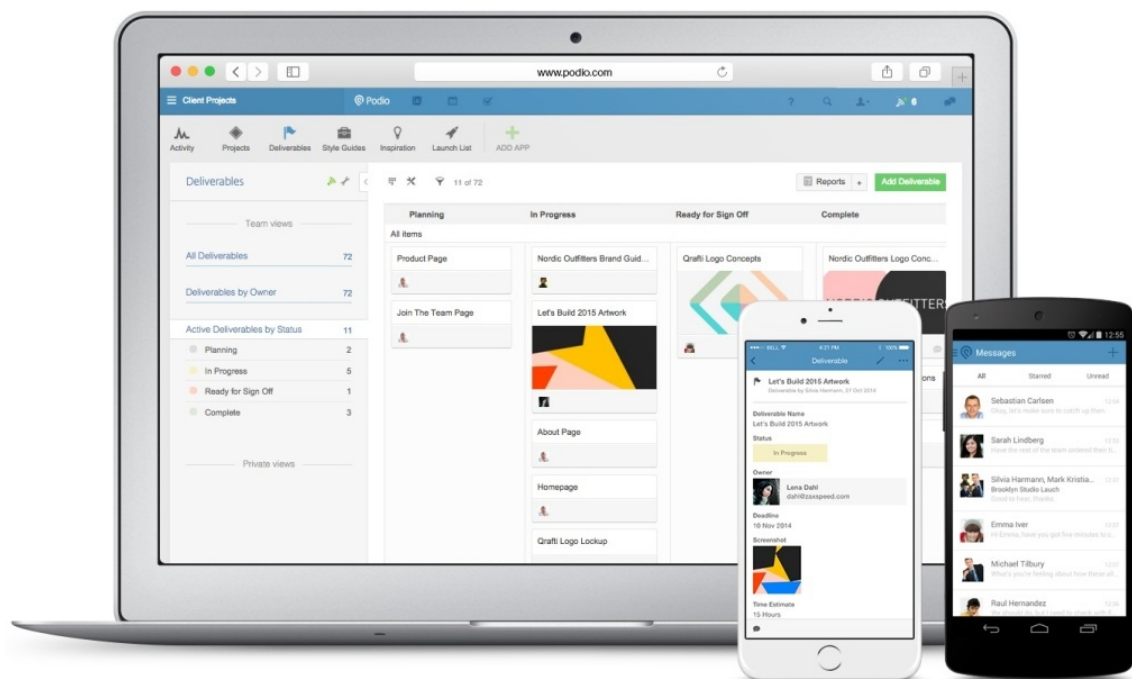
3.2.2 Podio

Podio je rozsáhlou aplikací, která nabízí zadávání úkolů v rámci týmu, možnost plánování schůzek, odevzdávání dokumentací k projektu a nahrávání jiných souborů, ankety, komentáře a mnohé další. Aplikace tedy může jednoduše sloužit jako jediná platforma používaná v souvislosti s daným projektem – bez nutnosti používat jiný komunikační kanál. Příklad obrazovky z aplikace Podio je zobrazen na obrázku číslo 3.2.

Podio je připraveno na práci na počítači i mobilních zařízeních a nabízí možnost zasílání upozornění na e-mail. Nabízí možnost opakování úkolů, tedy situaci, kdy se daný úkol pravidelně opakuje v týdenním, měsíčním či jiném intervalu.

Nevýhodou Podia je jeho zpoplatnění pro týmy větší než pět lidí. Zároveň nemusí být jeho používání vždy zcela intuitivní, a tak může uživateli trvat delší dobu, než se s aplikací seznámí a dovede ji plně využívat. Stejně jako Asana je Podio orientováno pouze na výstup a nikoliv na proces samotný, proto znovu umožňuje pouze dva stavy úkolu, tedy „splněno“ a „nesplněno“. Vzhledem ke komplexnosti Podia je obsah textu na úvodní stránce vysoký a zároveň nevyužívá příliš mnoho grafického zobrazení, což aplikaci ubírá na přehlednosti. [3]

¹Typ nástěnky, na níž se do sekcí „připravováno“, „v procesu“ a „hotovo“ přiřazují jednotlivé úkoly. Přesouváním jednotlivých úkolů napříč sloupci je možno zobrazit postup na projektu.



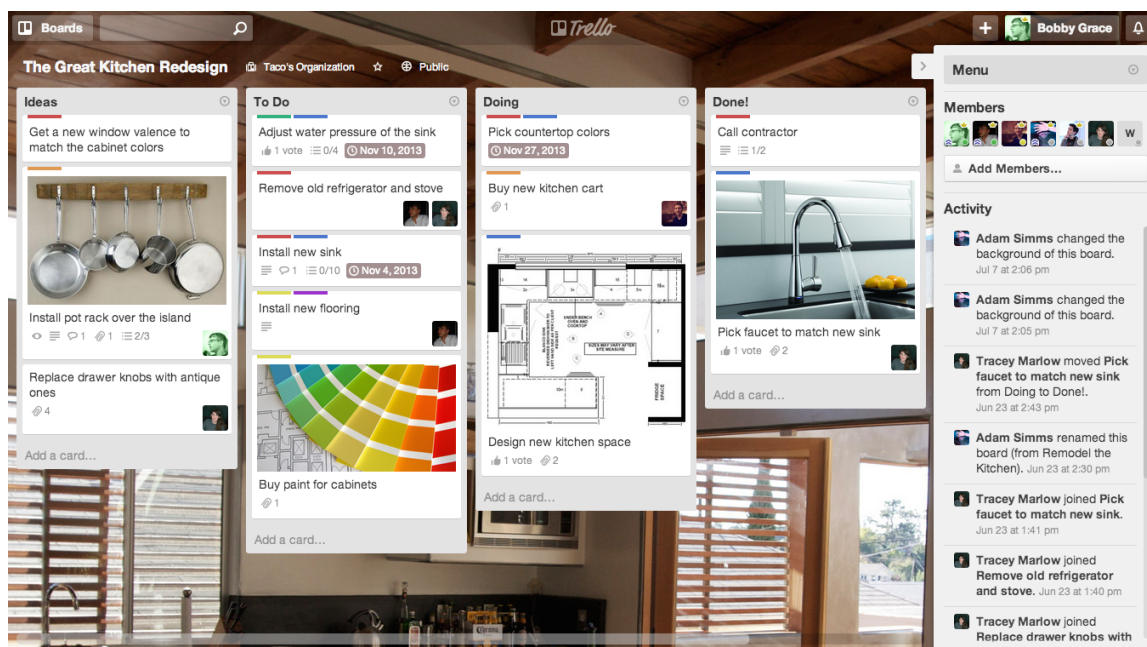
Obrázek 3.2: Úkoly konkrétního uživatele v aplikaci Podio, včetně zobrazení na mobilních zařízeních. [3]

3.2.3 Trello

Trello je velice jednoduchá aplikace, která umožňuje project management jednoduchým systémem malých kartiček ve stylu Kanban. Ty si uživatel „připíná“ do přednastavených sloupců, které značí stav projektu od prvotního nápadu přes „pracuji na tom“, až po „hotovo“. Sloupce se dají přizpůsobit – přidat i ubrat dle aktuální potřeby uživatele. Každá kartička s úkolem se dá dále sdílet s vybranými uživateli, či dokonce rozkliknout a přidat do ní dílčí úkoly. Tím se celkový postup v úkolu rozdělí na splněná procenta dle počtu podúkolu. Obrázek číslo 3.3 názorně ukazuje využití jednotlivých sloupců znázorňujících různý stav rozpracování projektu.

Stejně jako další aplikace je Trello propojeno s dalšími aplikacemi, jako je Google Calendar, Drive, Dropbox a další. Trello nabízí mimo desktopovou aplikaci také nativní mobilní aplikaci. Trello také nabízí základní funkcionalitu zdarma, která pro základní project management stačí. Dále pak nabízí pokročilejší funkce prostřednictvím placené verze Trello Gold (rozšíření pro jednotlivce), Trello Business (rozšíření pro celou skupinu uživatelů) a Trello Enterprise (rozšíření Trello Business). Trello dále umožňuje duplikaci projektů, čímž je usnadněno obnovení úkolu, který se děje opakovaně. Použití nástěnky typu Kanban umožňuje Trello hluboké přizpůsobení procesu uživateli, jelikož každý projekt může mít jiný průběh. Je tedy potřeba si přednastavené sloupce nástěnky personalizovat a přizpůsobit přesně danému projektu. Díky této variantě nabízí Trello také částečně funkci mood-trackeru, kdy jsou místo sloupečků progresu vyobrazeny jednotlivé nálady či rozpoložení uživatele, a uživatel pak přesouvá kartičku se svým jménem do sloupečku, který mu nejlépe vyhovuje. Není to sice optimální, přesto je to možné. Tato funkce je výhodná zejména pro osobnější týmy které pracují kompaktně a jsou založeny na interakcích mezi členy týmu.

Jistým nedostatkem aplikace je pak z mého pohledu horší sledování postupu na daném projektu. Ne vždy jeden ze čtyř podúkolů odpovídá 25 % odvedené práce. Zároveň se mohou jednotlivé nástěnky stát poněkud chaotickými, je-li na nich připnuto více kartiček s úkoly. Motivační funkce zde není zcela patrná, jelikož jsou nástěnky plné dalších kartiček jiných uživatelů, kteří na daném projektu spolupracují, a nejsou nikterak ucelené.[5]



Obrázek 3.3: Nástěnka v aplikaci Trello. Obsahuje seznam sloupců plných karet využívaných týmem nebo jednotlivcem [5]

3.2.4 Srovnání

Každá z výše zmíněných aplikací má jiné vlastnosti a může tak být vhodná pro jiné typy projektů. Pro srovnání a hodnocení jsem vytyčil jednotlivé kategorie požadavků tak, že jsem vybral několik hlavních cílových skupin mé aplikace (viz podkapitola 4.1.3). Vybranými cílovými skupinami jsou studenti pracující společně na zadaném projektu, dále menší firma pracující na běžných záležitostech a naposledy vyučující a jeho tým studentů pracujících na projektu.

Studenti pracující na zadaném projektu vyžadují zejména stejná editační práva pro všechny členy skupiny, jasné grafické vyjádření procesu, viditelné označení deadlinu a nízkou či nulovou cenu aplikace.

Menší firma pracující na mnoha projektech současně vyžaduje možnost kontroly zaměstnanců, limitovaná přístupová práva, jednoduché opakování úkolů. Cena je pro ně důležitá, nicméně si v některých případech mohou být ochotni připlatit za prémiové služby v aplikaci. Jednotlivé úkoly mohou být přiděleny jednotlivci či celé skupině osob.

Vyučující pracující na výzkumném projektu se skupinou studentů a dalších kolegů potřebuje zejména podpořit týmovou motivaci, sledovat náladu a stav týmu a zároveň připomínat členům týmu, aby v aplikaci vyznačovali postup na projektu pravidelně před týmovou schůzkou. Jednotlivé projekty jsou zpravidla přiděleny jednotlivým členům týmu.

Pro vyhodnocení srovnání jsem vybral základní požadavky každé z cílových skupin a hodnotil následně ve všech třech aplikacích všechna kritéria na škále 0 až 3, jak je zobrazeno v tabulce č. 3.1. Výstupy srovnání jsem rozpracoval i v textové podobě s přidáním závěru pro využití ve vyvíjené aplikaci.

Požadavek	Asana	Podio	Trello
Práva k editaci	+	+++	+++
Jasný přehled postupu	+		++
Možnosti deadlinů	+++	+++	+++
Cena	++	+	++
Duplikace projektu	+++	+++	+++
Přehlednost při více úkolech	++	++	+
Nálada a subjektivní ukazatele			+
Motivační prvky			+
Upomínka k aktualizaci postupu	++	+++	+++

Tabulka 3.1: Srovnání jednotlivých existujících řešení

- Práva k editaci – Asana jako jediná nabízí pouze variantu přidělení jednoho úkolu jednomu uživateli, zatímco Podio i Trello nabízejí možnost přiřadit k jednomu projektu více uživatelů. Pro naši aplikaci by měla existovat možnost přiřadit k projektu více osob, avšak pro jednotlivé podúkoly by měl existovat pouze jeden zodpovědný člen týmu, případně pak může podúkol zůstat nepřidělen nikomu, kdy jej mohou editovat všichni členové týmu.
- Jasný přehled postupu – Ani jedna z aplikací nenabízí jasné grafické vyobrazení postupu na úkolu. Asana i Podio nabízejí pouze variantu „splněno“ a „nesplněno“, přičemž Asana umožňuje alespoň formu Kanban nástěnky a sledování grafu zadaných a splněných úkolů. Trello pak postup zobrazuje ukazatelem, který však projekt dělí přesně dle počtu podúkolů a neumožňuje nastavit jinou váhu jednotlivým podúkoly. Jednoznačný přehled postupu byl součástí zadání mé diplomové práce, proto bude projekt vyvíjen tak, aby bylo možno i z dálky zhodnotit postup na úkolu.
- Možnosti deadlinů – Všechny aplikace nabízejí možnost zadání finálního data dokončení projektu a přehledně jej pak u projektů vypisují. Podobným způsobem budou použity deadliny také v mnou vyvíjené aplikaci.
- Cena – Asana a Podio limitují své bezplatné verze pro nižší počet uživatelů, Asana podporuje zdarma maximálně 15 uživatelů ve skupině, Podio maximálně 5. Trello má základní verzi zdarma s možností dokoupení Gold verze, která se cenově blíží ceně rozšířené verze Asany.
- Duplikace projektu – Asana, Podio i Trello podporují možnost duplikace projektu v situacích, kdy se některý úkol opakuje pravidelně. Po vzoru těchto bych rád implementoval možnost duplikace projektů také.
- Přehlednost při více úkolech – Díky seznamovité struktuře jsou zde ve výhodě Asana a Podio, které jednotlivé úkoly řadí pod sebe do seznamu na základě jejich deadlinu. Trello se může zdát při větším množství úkolů zmatené a nejasné. Rád bych připravil

přehlednou nástěnku, na níž bude jasně graficky znázorněn stav všech částí projektu tak, abych předcházela zmatenosti a zbytečnému textu okolo.

- Nálada a subjektivní ukazatele – S výjimkou Trelly je prakticky nemožné použít aplikace k zaznamenání subjektivních ukazatelů, jako jsou nálada či emoce. Z tohoto důvodu plánuji aplikovat systém ukazatelů, které může uživatel jednoduše měnit oběma směry.
- Motivační prvky – Vesměš žádná z aplikací nemotivuje uživatele k zaznamenávání postupu či srovnávání se s dalšími členy týmu. Trello jako jediné nabízí grafickou možnost srovnání, která však není zcela jasná. Pro účely této diplomové práce jsem se rozhodl nebrat v potaz jednorozce, které nabízí Asana. Věřím, že jasné grafické zobrazení všech částí projektu nabídne uživateli jednoduché srovnání s ostatními uživateli, zejména v případě více studentů pracujících na podobném zadání.
- Upomínka k aktualizaci postupu – Všechny zmíněné aplikace nabízejí možnost upomínky uživatele skrze e-mail či notifikaci v aplikaci. Asana však v základním nastavení posílá upomínek příliš mnoho. Rád bych se proto poučil z Asany a upozornění zasílal v nižší frekvenci a zároveň umožnil uživateli přizpůsobit si upomínky – jejich text, frekvenci i kterým uživatelům se budou zasílat, aby tak zbytečně neobtěžovaly některé uživatele, kterých se netýkají.

Ze srovnání jsem vyhodnotil, že pro studenty pracující na společném projektu by bylo nejvýhodnější aplikací Trello, které má nejlepší vlastnosti v kritériích přidělování úkolů, jasného přehledu postupu a jednoduše kontrolovatelných deadlinů. Pro tuto skupinu v aplikaci však chybí jasný grafický způsob hodnocení postupu a případně také motivační prvky.

Pro malou firmu bych z nabízených aplikací vybral Asanu, která je příznivá cenou, nabízí jednoduchou možnost duplikace projektů a v placené verzi nabízí dokonce i šablony. Zároveň je relativně přehledná i při větším počtu úkolů. Všechny tři aplikace mají relativně vysoký průnik a proto si myslím, že bude menší firma schopna fungovat s jakoukoliv z nabízených aplikací.

Nejlépe z hodnocených aplikací pro vyučujícího a jeho výzkumný tým vyšlo Trello, ač jsem jej hodnotil podprůměrně ve dvou ze tří kritérií. Trello skvěle zvládá připomenutí uživatelům, bohužel si však nevede tak dobře, přijde-li na motivační prvky a dostatečnou variabilitu ukazatelů na to, aby mohly být použity i pro náladu či subjektivní pocity. Tato uživatelská kategorie měla celkově nejmenší průnik s nabízenými aplikacemi, proto by bylo vhodné se v návrhu vlastní aplikace soustředit také na pokrytí potřeb této skupiny a jí podobných.

Srovnání zároveň ukázalo, že všechny tři aplikace mají spíše delší učicí křivku, proto nemusí být vhodné pro uživatele pracující na mikropojektech či uživatele, pro které jsou stávající aplikace příliš složité a nemají čas ani chuť se učit zacházet s novou aplikací. V těchto případech je sledování postupu žádané, musí však být naprosto jednoduché, intuitivní a nesmí zabrat více než pár vteřin.

Kapitola 4

Návrh aplikace

Zadáním tohoto projektu bylo vytvořit webovou aplikaci dostupnou uživatelům z webového serveru přes počítačovou síť Internet. V rámci této aplikace by uživatel měl být schopný zaznamenávat postup na svých projektech či záležitostech vizuálním způsobem. Zároveň by měl být schopný sledovat postup svých kolegů, podřízených či spolužáků.

Způsob zaznamenávání uživatelského postupu by měl být pokud možno co nejjednodušší a měl by jej zvládnout i technicky nepříliš zdatný jedinec. Celá aplikace by zároveň měla klást důraz na vizuální reprezentaci stavu jednotlivých úkolů a měla by být silně uživatelsky zaměřena, jednoduchá na používání i pochopení. Z každého záznamu uživatelského postupu by mělo být jasné patrné, co nového na projektu udělal, a v jaké fázi rozpracovanosti projektu se právě nachází.

V první sekci této kapitoly popisují vlastnosti, které by měla vyvíjená aplikace mít, a způsob jakým jich dosáhnout. V druhé sekci číslo 4.2 pak navrhuji způsob, jakým tyto vlastnosti co nejlépe reprezentovat v prostředí webové aplikace.

4.1 Návrh systému

4.1.1 Základní prvky systému

Pro jednoduché zaznamenávání uživatelského postupu je vhodné jím řešenou záležitost rozdělit na menší logické celky tak, aby bylo pro uživatele snadné zaznamenávat a sledovat i záležitosti většího rozsahu. Na základě této úvahy jsem se rozhodl celý systém rozdělit do těchto tří logických celků:

Projekt – je abstrakcí celé uživatelské záležitosti nebo projektu, u kterého si přeje sledovat průběžný stav dokončení. Skládá se z jedné nebo více záležitostí. Celý projekt by mělo jít snadno sdílet s ostatními spolupracovníky, a to jak hromadně – s více uživateli najednou, tak i s každým uživatelem individuálně.

Každý uživatel se může účastnit práce na jednom i více projektech.

Záležitost – nebo také anglicky *issue*, je abstrakcí menší skupiny společných úkolů. Každá záležitost náleží do daného projektu a může se skládat z jednoho až čtyř úkolů.

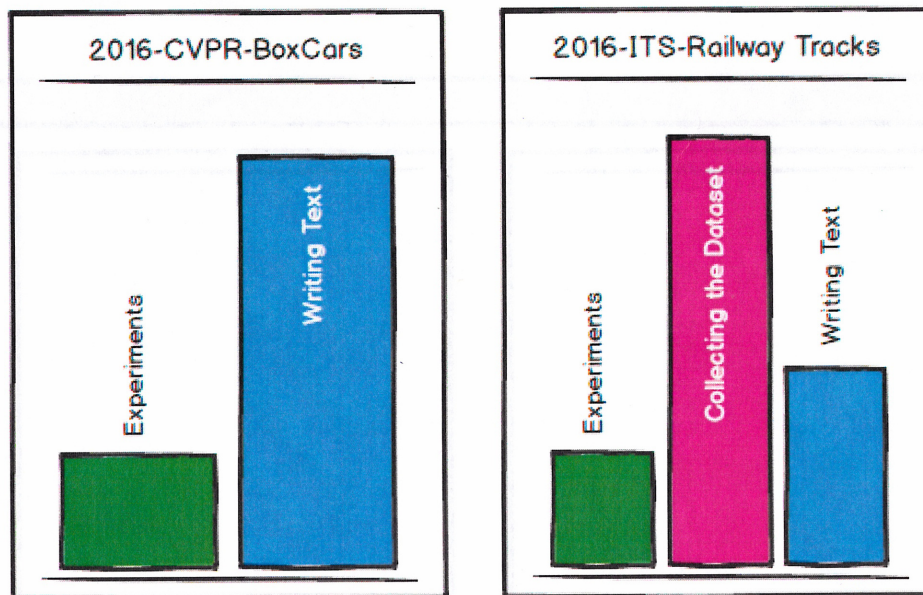
U záležitosti existují dvě varianty. Buď ji má celou na starost pouze jeden pracovník, nebo do ní může přispívat více spolupracovníků najednou. Stejně jako projekt by ji mělo jít jednoduše sdílet buď se skupinou lidí, nebo s každým uživatelem individuálně.

Úkol – nebo také anglicky *task* či *gauge* (od slova *měřidlo*, kterým jsou v aplikaci tyto úkoly reprezentovány) je nejmenší logická jednotka této struktury a je také jediná, která nese informaci o samotném stavu daného úkolu a tím také o stavu celého projektu.

Každý úkol by měl nést informaci o tom, v jakém stavu se nachází. V rámci co největší jednoduchosti celé aplikace (a na rozdíl od některých dalších aplikací pro sledování projektů, kde je stav úkolu často reprezentován i textově), je stav každého úkolu reprezentován skalární veličinou, konkrétně číslem od 0 do 100 procent. Nejčastěji bude nula znamenat „nový, nezapočatý úkol“, zatímco hodnota sto pak bude reprezentovat již zcela dokončený úkol. V aplikaci ale není záměrně této hodnotě přiřazován žádný další název a procentuální hodnota úkolu může také reprezentovat například spokojenost uživatele s kvalitou splnění celé záležitosti, nebo třeba jeho aktuální náladu.

Stav úkolu by zároveň bylo dobré uživateli prezentovat ve vhodné vizuální podobě, jakou může být například sloupcový graf. Z toho lze již z dálky minimálně odhadnout, v jakém stavu se úkol nachází (jestli se pohybuje například okolo 0 %, 50 %, nebo 100 %), a této vlastnosti lze například využít i při porovnávání více podobných záležitostí, které se například skládají ze stejných úkolů, ale jsou přiřazeny rozdílným uživatelům.

Jeden úkol náleží právě jedné záležitosti a každá záležitost může mít maximálně čtyři úkoly. Toto číslo bylo navrženo tak, aby i při pohledu na více záležitostí najednou byla každá z těchto záležitostí dostatečně přehledná, a zároveň aby byl uživatel donucen přemýšlet nad strukturou svého projektu a například nepřiradil všechny úkoly pouze k jedné záležitosti. V závislosti na zpětné vazbě od uživatelů při testování aplikace se dá tato hodnota upravit.



Obrázek 4.1: Návrh reprezentace stavu několika různých úkolů výškou grafu, který byl součástí zadání této práce

Každý úkol by měl mít na starost právě jeden uživatel, jelikož při takto navržené struktuře nedává smysl, aby na něm pracovalo více uživatelů současně – samotná jeho hodnota může být velmi subjektivní a každý uživatel ji může vnímat mírně jinak.

Na obrázku číslo 4.1 je příklad návrhu reprezentace několika různých úkolů. Každý úkol má pro jednodušší rozlišitelnost vlastní barvu a název a výška jeho grafu poté reprezentuje jeho procentuální hodnotu.

4.1.2 Základní vlastnosti systému

Navržená aplikace by měla odpovídat principům uživatelského návrhu popsaným v kapitole číslo 2.2. Musí být jednoduchá na používání i pochopení pro případ, že by chtěl uživatel spolupracovat na malém projektu se svými známými, kteří tuto aplikaci nikdy předtím neviděli a nemají ji v plánu sami používat, tedy investovat čas do pochopení aplikace. Vše musí jít jednak jednoduše sdílet, a zároveň by měli nově pozvaní uživatelé ihned pochopit, o co se jedná, co se po nich v aplikaci chce a jak toho dosáhnout.

Z tohoto plyne, že by aplikace měla umožňovat editaci i nepřihlášeným uživatelům (pokud si to majitel projektu v nastavení zvolí), ale měla by dokázat rozlišovat mezi jednotlivými nepřihlášenými uživateli. Zároveň by měla poskytnout způsob přihlášení a načtení uživatelských dat z různých prohlížečů.

Aplikace musí vypadat jednotně a být stejně použitelná ve všech aktuálních verzích moderních webových prohlížečů. Zároveň by se měla dokázat přizpůsobit i zařízením s dotykovou obrazovkou. Ideálně je potřeba už při návrhu komunikačního protokolu mezi klientem a serverem myslet na možné rozšíření o nativní mobilní aplikaci, kterou by uživatelé mohli používat i v off-line módu na cestách.

4.1.3 Cílová skupina

Při návrhu aplikace bylo identifikováno několik možných cílových skupin. Jednotlivé případy byly zmapovány a brány v potaz při vývoji.

Skupina studentů – může se jednat o menší skupinu spolužáků řešících společný projekt v rámci výuky. Tito studenti by potřebovali vědět termíny, do kterých mají úkol dokončit, a také přesné rozdělení úkolů v rámci týmu. Pro vedoucího týmu by bylo vhodné vědět jak jsou na tom ostatní členové a případně kdo z týmu svou práci nestíhá a je potřeba mu pomoci.

Vyučující vedoucí studenty – například vyučující vedoucí tým studentů magisterského nebo doktorského studia. V tomto případě vyučující potřebuje přehled o aktivitě svých studentů a o jejich výsledcích i mimo konzultační hodiny. Pokud studenti pracují v rámci stejné skupiny, bylo by pro ně dobré vidět i postup svých spolužáků tak, aby jej mohli srovnat se svým vlastním a věděli tak, kterým oblastem je potřeba věnovat se přednostně.

Pokud zároveň tento tým pořádá pravidelné schůzky, hodila by se zde funkce připomenout studentům aktualizovat své změny a zobrazit, co se od poslední schůze udělalo.

Zájmová organizace – menší skupina lidí (například okolo 20 členů) organizující společné akce. Jejich akce jsou limitovány přesným termínem a často se opakují (například

organizace společenské akce nebo letního tábora). Organizace se skládá z menších týmů, které často mají ještě vlastního vedoucího.

Zde je nutné, aby vedoucí každého týmu mohl kontrolovat stav úkolů svých podřízených, ale zároveň sám dokázal interpretovat postup svému vedoucímu. Organizace také často pořádá pravidelné schůze, kde se hodnotí dokončené úkoly a plánují se budoucí akce.

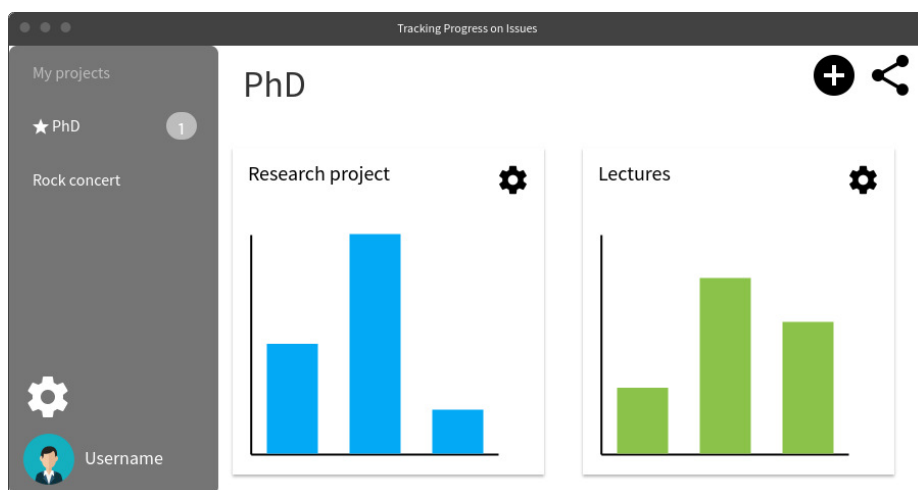
Malý tým v rámci firmy – strukturou i pojetím velmi podobná skupina jako nezisková organizace, ale je zde kladen větší důraz na uživatelské role a práva. Firemní struktura má pevně stanoveného vedoucího a jeho podřízené. I v případě dokončení úkolu je zde často nutná ještě kontrola (potvrzení) vedoucího, že práce byla provedena správně a zaměstnanec za ni může být odměněn.

Druhou variantou je, že tým používá tuto aplikaci pouze jako doplněk k existujícím aplikacím popsaným v kapitole 3.2. V tomto případě firma využívá silných stránek těchto aplikací, jako je například podrobné sledování úkolů, chyb atd. Tuto aplikaci používá pro jednoduchou vizuální reprezentaci stavu vytvořených úkolů a možnost rychlého a snadného sledování pokroku.

Skupina kamarádů – uživatel pořádá menší akci se svými sousedy a přeje si mít přehled o zajištění všeho potřebného na pořádanou událost. Tento uživatel nebude používat žádné pokročilé funkce, ale naopak bude potřebovat jednoduše sdílet jím vytvořený projekt se svými sousedy, od kterých nebude očekávat složité přihlašování, ale pouze krátkou aktualizaci toho, co zařídili, dostupnou na jedno až dvě kliknutí v rámci vteřin.

4.2 Návrh uživatelského rozhraní

V této sekci je popsán způsob, jakým byly navrženy nejdůležitější prvky grafického uživatelského rozhraní, a které vlastnosti by měly dle mého názoru splňovat.



Obrázek 4.2: Počáteční návrh rozložení obrazovky a uživatelského rozhraní

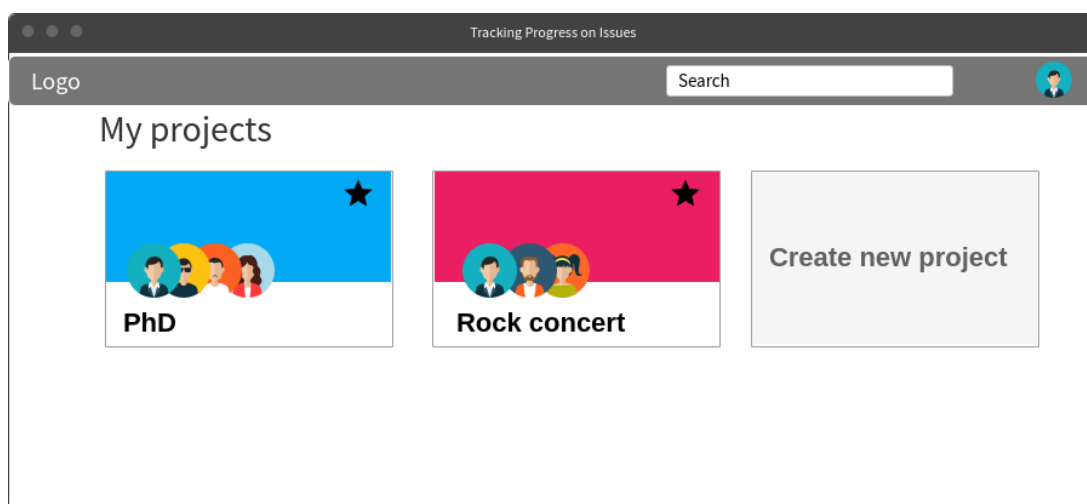
Na obrázku číslo 4.2 je zobrazen jeden z prvotních návrhů uživatelského rozhraní a způsobu rozložení ovládacích prvků. Tento návrh byl však ještě mnohokrát přepracován a od konečné podoby aplikace je dost vzdálen.

4.2.1 Nástěnka uživatele

Nástěnka uživatele by měla být první stránka, která se otevře uživateli po načtení této aplikace. Je-li uživatel v aplikaci zcela nový a není zatím součástí žádného projektu, zobrazí se mu stránka s představením aplikace popsána v podkapitole Titulní strana.

Na této nástěnce by měl být zobrazen přehled všech uživatelských projektů a měla by sloužit jako rychlý rozcestník, ze kterého dokáže uživatel na jedno kliknutí spravovat všechny své důležité úkoly a projekty. Přednostně jsou na nástěnce zobrazeny oblíbené projekty a u každého z projektů je zde možnost přiřadit ho k oblíbeným, stejně jako možnost vytvořit projekt nový. Inspirací pro toto uspořádání nástěnek byly i příklady z reálných produktů, jako je například Trello [5]. Dále by tato stránka měla zobrazovat seznam uživatelem archivovaných či jinak uzavřených projektů.

Na obrázku číslo 4.3 je vidět prvotní návrh uživatelské nástěnky. Takto navržená stránka slouží pouze jako rozcestník mezi uživatelskými projekty a neukazuje žádné aktuální informace. Ty byly přidány až v pozdější fázi návrhu. Při tomto návrhu jsem se inspiroval prvky stylu *Material design* od společnosti Google, ale například i rozložením nástěnky aplikace Trello.



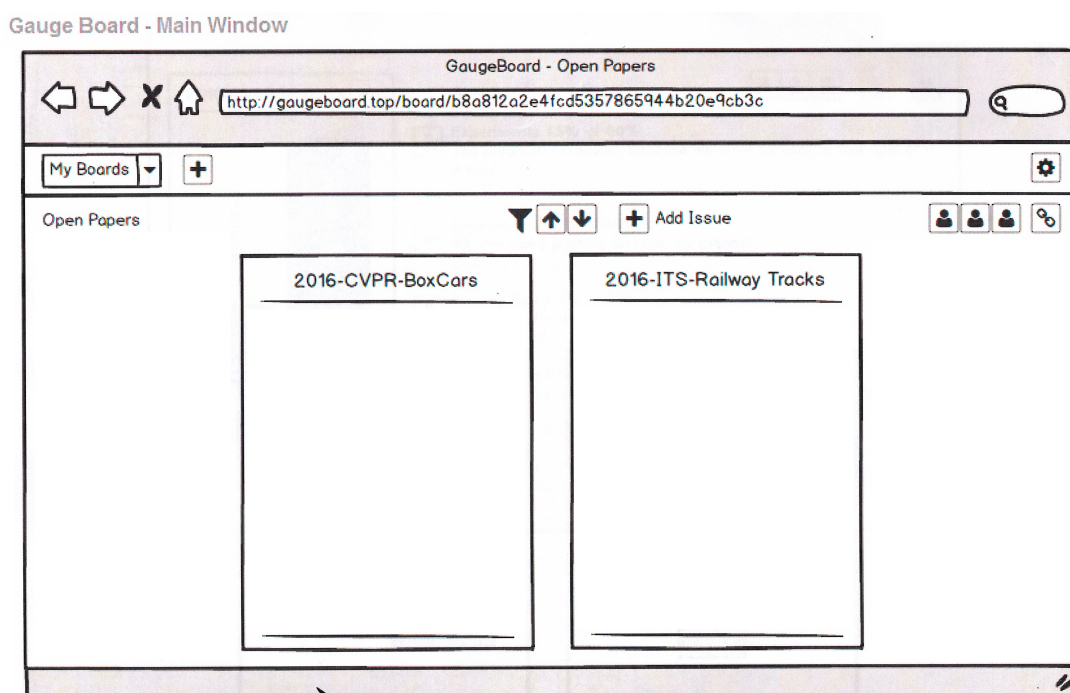
Obrázek 4.3: Prvotní návrh rozložení uživatelské nástěnky se seznamem jeho projektů

Na základě zpětné vazby od uživatelů během testování aplikace (konkrétně bodu 6.2.2) byl na tuto nástěnku také přidán přehled všech uživatelských termínů seřazených podle termínu dokončení (od nejnaléhavějších po ty nejvzdálenější) tak, aby uživatel ihned po přihlášení do aplikace dokázal určit, který úkol by měl dostat prioritu a na kterém začít pracovat nejdříve.

4.2.2 Nástěnka projektu

Nástěnka projektu je jeho středobodem a rozcestníkem pro všechny přispěvatele v daném projektu. Jejím hlavním úkolem je zobrazovat seznam všech záležitostí v projektu, včetně jejich aktuálního stavu. Z nástěnky tak může uživatel kliknutím přejít na vybranou záležitost a v ní aktualizovat svůj postup. Může také nástěnku využít k porovnání výsledků (hodnot jednotlivých grafů) uživatelů, což se může hodit jak vedoucímu projektu, který jednoduše identifikuje části projektu, které musí věnovat více pozornosti, tak i jednotlivým účastníkům projektu, kteří mohou porovnat svůj postup s ostatními a tím nabývat motivaci.

Na obrázku číslo 4.4 je zobrazen prvotní návrh této obrazovky, který byl součástí zadání práce. Tento návrh počítá se zobrazením karet všech záležitostí přidáných v projektu. Vedle toho je zde možnost projekt editovat, přidat novou záležitost, spravovat uživatelská práva, projekt sdílet aj.

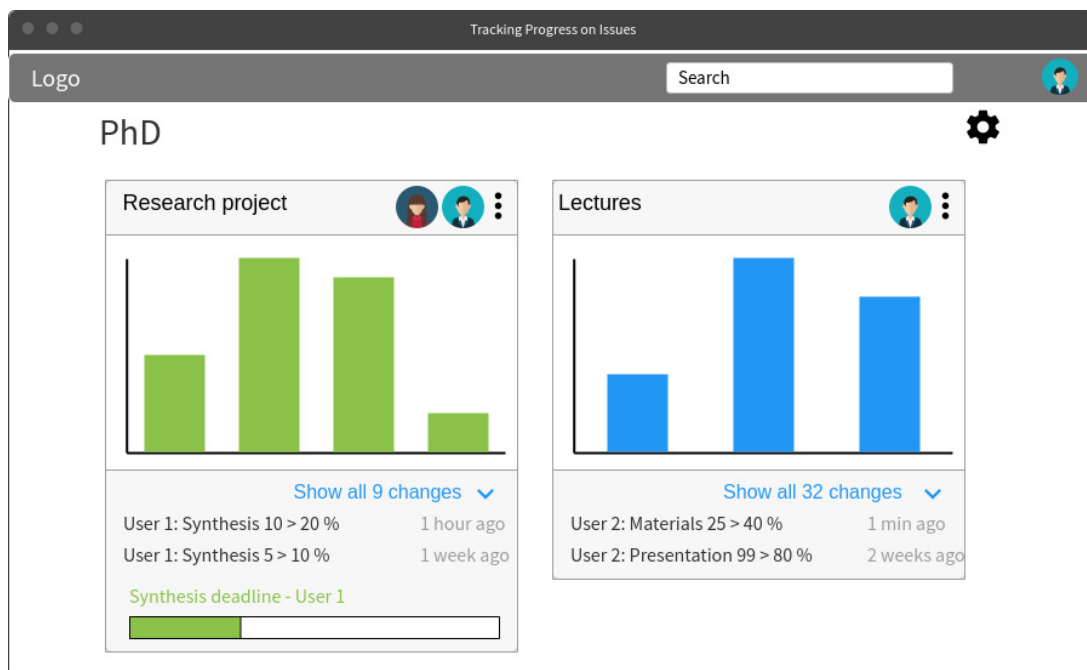


Obrázek 4.4: Návrh obsahu a rozložení nástěnky projektu, který byl součástí zadání práce

V návrhu na obrázku číslo 4.5 jsem se zaměřil na karty záležitostí. Jejich primárním účelem je jednoduchým způsobem poskytnout uživateli všechny důležité aktuální informace. Je zde tedy graf s aktuálním stavem všech úkolů v záležitosti a to tím způsobem, aby bylo i z dálky patrné, v jakém stavu se úkol nachází, a případně se dal porovnávat s ostatními úkoly. Dále je zde výpis posledních dvou změn v záležitosti včetně data provedené této změny, takže je ihned patrné, kdy uživatel na úkolu pracoval naposledy. Pod seznamem posledních změn byl v určité fázi návrhu i seznam všech termínů, to se však při testování ukázalo jako zbytečný a rušivý prvek a tento seznam byl později odstraněn.

Součástí uživatelského rozhraní u karty záležitosti je i seznam uživatelů, kteří se na úkolu podílí, což pomůže lepší asociaci konkrétní osoby s danou záležitostí v případě, kdy více lidí nezávisle na sobě pracuje na podobně znějícím úkolu. V neposlední řadě by součástí

karty mohl být i seznam rychlých akcí, který by pomohl zejména správci projektu při jeho nastavování (obsahoval by například rychlý přístup ke sdílení záležitosti, nastavování uživatelských práv, mazání, atd.)



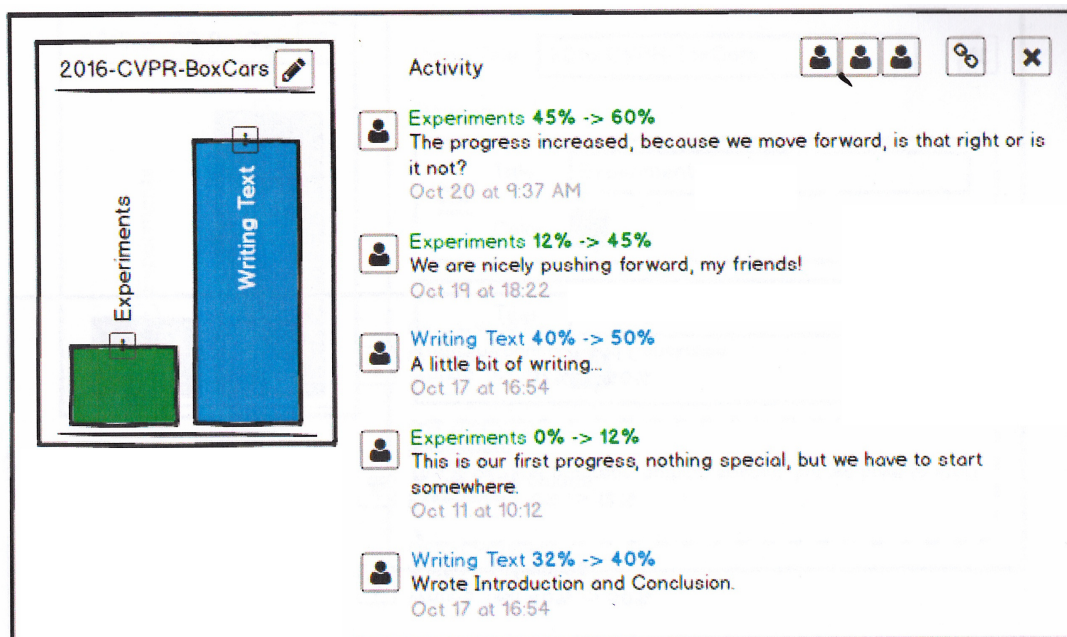
Obrázek 4.5: Návrh rozložení obrazovky jednoho konkrétního projektu

Součástí nástěnky projektu by také měla být možnost projekt editovat, spravovat uživatelská práva k projektu, sdílet jej a nebo jej smazat. Kromě operace mazání by měl mít správce projektu možnost projekt uzavřít i jiným způsobem nazvaným archivace. Touto operací by se projekt již nezobrazoval v uživatelském seznamu aktivních projektů (na stránce Nástěnka uživatele), ale byl by umístěn do seznamu „archivovaných projektů“. Zároveň by již projekt ani žádný z úkolů neumožňoval provádět žádné změny, ale byl by stále dostupný všem svým uživatelům k prohlížení. V poslední řadě by měl mít uživatel bez administrátorských práv možnost projekt opustit, čímž by sám sobě odebral všechna přístupová práva a projekt by se mu již nikde nezobrazoval.

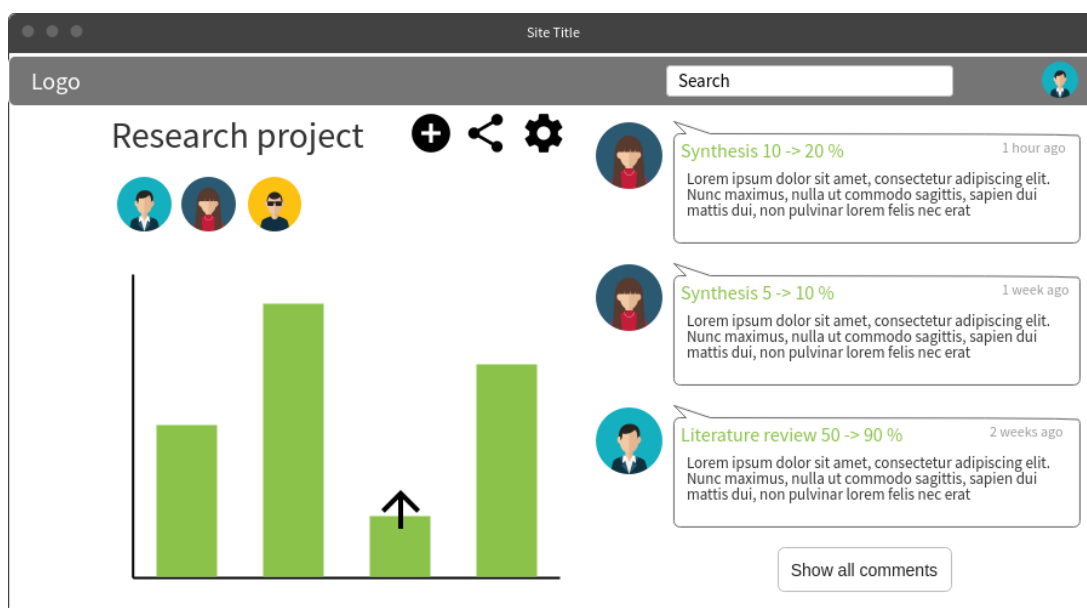
4.2.3 Detail záležitosti

Tato stránka obsahuje nejdůležitější součást celé aplikace – samotný způsob zaznamenávání úkolů a dá se tedy očekávat, že zde uživatel stráví nejvíce času z celé aplikace. Tato stránka by měla umožňovat měnit stav jednotlivých úkolů, prohlížet všechny změny úkolů, editovat úkoly aj.

Obrázek číslo 4.6 ukazuje základní myšlenku této obrazovky, která byla součástí zadání. Na její levé straně je graf s hodnotou jednotlivých úkolů. Těm lze táhnutím myši měnit jejich výšku, a tím i aktualizovat jejich hodnotu. Na pravé straně obrazovky je seznam změn seřazený od nejnovějších k nejstarším. Součástí tohoto výpisu je ikona uživatele, který změnu provedl, název, stará a nová hodnota grafu a uživatelův textový popis provedené změny, uvede-li jej.



Obrázek 4.6: Návrh obrazovky s detailem jedné záležitosti, který byl součástí zadání práce



Obrázek 4.7: Upravený návrh rozložení obrazovky s detailem jedné záležitosti, který poskytuje větší prostor pro graf s úkoly i pro komentáře ke změnám úkolů

Na obrázku číslo 4.7 je přepracovaný návrh vzhledu této stránky. Ta je rozdělena na dvě rovnoměrné poloviny. V pravé polovině obrazovky zůstává v principu stejný výpis provedených změn, u kterého se pouze počítá s přesunutím data provedené změny na řádek spolu

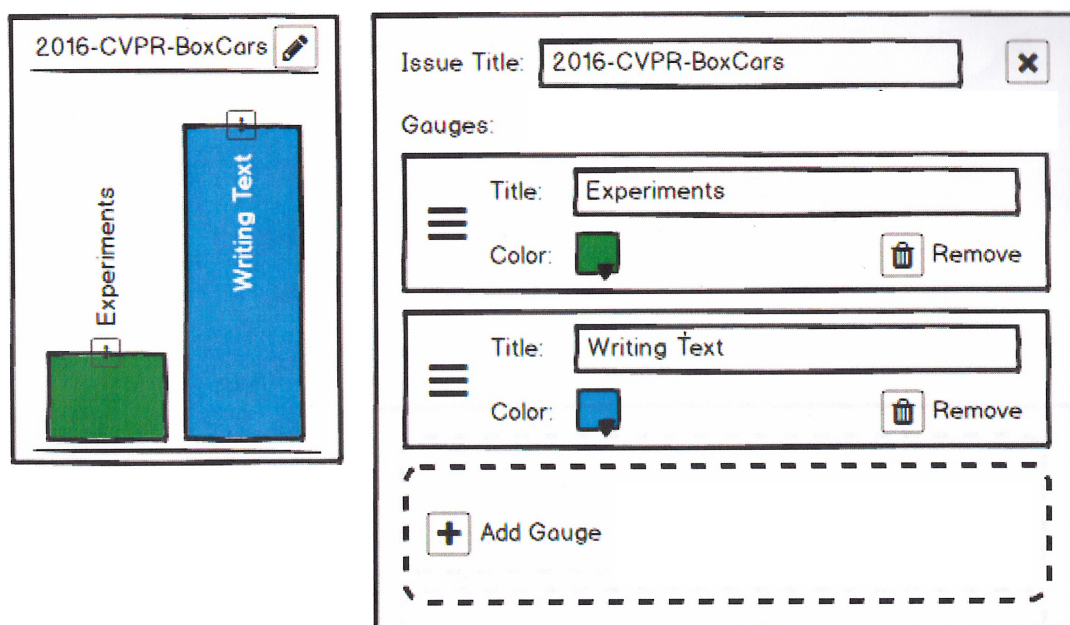
s názvem úkolu, čímž lze ušetřit trochu prostoru na výšku tak, aby se na obrazovku vešlo více komentářů.

Všechny ovládací prvky jsou však přesunuty na levou stranu obrazovky tak, aby bylo vytvořeno co nejvíce prostoru pro seznam změn, u kterého se počítá s tím, že může být u pokročilých projektů relativně dlouhý. Mezi ovládacími prvky by měla být možnost vytvořit nový úkol, měnit hodnoty stávajících úkolů i celé záležitosti a možnost spravovat uživatelská práva k záležitosti.

4.2.4 Úprava záležitosti

Po kliknutí na ikonu ozubeného kola v obrazovce s detailem záležitosti by se uživateli měla v pravé části okna objevit možnost upravovat danou záležitost.

Uživatel by zde měl mít možnost změnit název jednotlivých úkolů, jejich barvu, jejich pořadí v grafu a případně je i smazat. Možnost měnit pořadí jednotlivých úkolů byla navržena i na základě studie nejčastějších požadavků při testování v bakalářské práci Silvie Janíkové [14].



Obrázek 4.8: Návrh obrazovky s úpravou záležitosti a jednotlivých úkolů dodaný spolu se zadáním této práce

Na obrázku 4.8 je vidět původní návrh této obrazovky připravený spolu se zadáním této práce. Oproti tomuto návrhu jsem se rozhodl přesunout úpravu záležitosti (v tomto případě pouze úpravy jejího názvu) na samostatnou stránku, na které lze měnit více parametrů záležitosti. Zároveň jsem se rozhodl dát dialog pro vytváření nového úkolu pod samostatné tlačítko tak, aby bylo pro uživatele ihned jasně viditelné.

4.2.5 Titulní strana

Titulní strana by se měla zobrazit novým uživatelům, kteří ještě v aplikaci nemají vytvořený účet, ani nejsou přidáni k žádnému z projektů. Jejím účelem je seznámit nové návštěvníky s aplikací a vysvětlit jim jak ji používat.

Pro tuto stranu jsem navrhl tři kreslené obrázky popisující tři základní prvky aplikace (tedy sledování postupu pomocí grafu, komentování vytvořených změn a vizuální reprezentaci několika záležitostí vedle sebe). K těmto obrázkům jsem také navrhl vytvoření dvou krátkých videí jednoduše zachycujících změny v aplikaci bez nutnosti dlouhých popisků.

Obrázek výsledné implementace titulní strany a obrázky vytvořené k seznámení uživatelů s aplikací jsou přiloženy v příloze C.1.

4.3 Druhá fáze návrhu

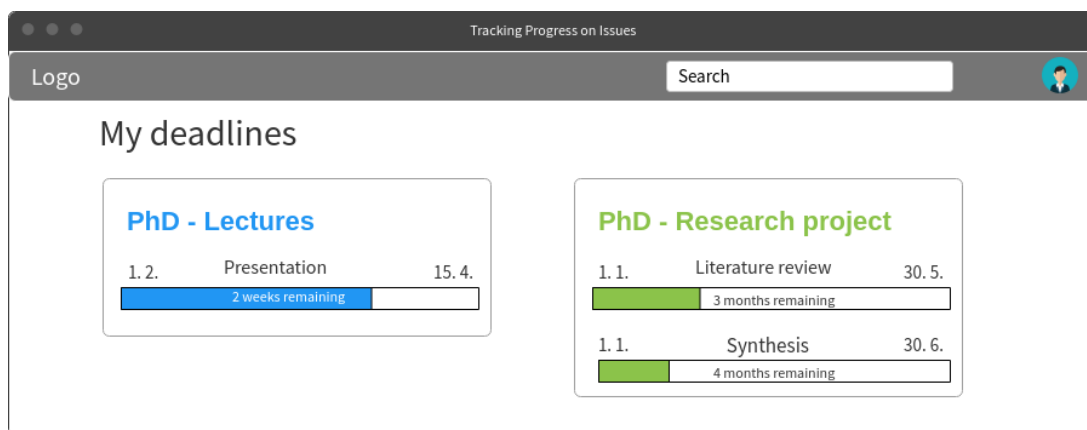
V této sekci je popsán návrh dalších struktur aplikace, které nebyly patrné hned ze začátku vývoje aplikace, ale jejichž nutnost vyplynula až z uživatelského testování (popsaného v kapitole 6.2) a na základě zpětné vazby od uživatelů používajících tuto aplikaci.

4.3.1 Termíny

Již v rané fázi testování (viz podkapitola 6.2.1) vyplynula nutnost mít v aplikaci možnost vytvářet a sledovat termíny (anglicky *deadline*) jednotlivých úkolů.

Každý úkol, ale i záležitost, by tak měly mít novou položku reprezentující termín dokončení. Ten by měl obsahovat datum začátku a konce tohoto termínu a případný komentář vysvětlující, co vše je potřeba na úkolu dokončit. Pro vizuální reprezentaci termínu jsem se rozhodl zvolit tzv. *progress bar*, neboli česky „ukazatel průběhu“. Z toho je již na první pohled patrné, kolik procent času uživateli zbývá, a lze do něj jednoduše přidat i textové pole s počtem dnů či týdnů, které do konce termínu zbývají.

Na obrazovce s detailem záležitosti navržené v podkapitole 4.2.3 jsem se pak rozhodl umístit tyto ukazatele pod graf s vizualizací úkolů, pod kterým je velké množství volného prostoru zejména v případě, kdy je u záležitosti větší množství komentářů a změn.



Obrázek 4.9: Návrh obrazovky s přehledem všech uživatelských termínů seskupených podle záležitostí, ke kterým náleží

Během testování (podkapitola číslo 6.2.2) také vznikl požadavek na navržení samostatné stránky s přehledem všech uživatelských termínů. Návrh této stránky je zobrazen na obrázku číslo 4.9. Na něm jsem se rozhodl sloučit všechny uživatelské termíny podle jednotlivých záležitostí do jedné skupiny a tyto skupiny jsou pak seřazeny podle data jejich ukončení od nejbližšího k nejvzdálenějšímu.

4.3.2 Duplikace projektu

Dalším z požadavků na návrh, který vyplynul z počáteční fáze testování, je možnost duplikovat projekt, což je funkce, která výrazně zjednodušuje zadávání podobných, často se opakujících projektů (více v podkapitole testování 6.2.4).

Možnost kopírování by měla být dostupná v nastavení projektu. Uživatel by měl být schopný zvolit název nového projektu a ideálně by si měl být schopný vybrat, které vlastnosti projektu si přeje kopírovat.

Složitější je ovšem kopírování termínů v projektu, protože nemá smysl kopírovat termíny rok starého projektu. Jediná možnost, která mne při návrhu napadla je dovolit uživateli vybrat si datum začátku projektu, od kterého budou všechny termíny začínat s tím, že jejich konec bude vypočítán na základě tohoto data začátku a jejich původní délky trvání. Pokud by si uživatel přál, aby některé z termínů začínaly v jiný den, bude muset jejich čas upravit individuálně.

Samotná duplikace pak probíhá následovně:

1. Vytvoření nového projektu s novým jménem.
2. Přiřazení nových přístupových práv k novému projektu podle práv uživatelů ke starému projektu
3. Pro všechny záležitosti patřící ke starému projektu:
 - (a) vytvoření nové záležitosti s názvem i nastavením původní záležitosti,
 - (b) přiřazení nových přístupových práv k nové záležitosti podle práv uživatelů k původní záležitosti,
 - (c) vytvoření nových úkolů se stejným jménem i nastavením, ale s hodnotou splnění 0 %,
 - (d) vytvoření nových termínů patřících ke stejným úkolům, ale s novým časem začátku (nastaveným uživatelem) a s časem konce vypočteným jako $novy_zacatek + cas_trvani_puvodniho_terminu$.
4. Zkopírování nastavení připomenutí původního projektu k novému projektu.
5. Informování všech zúčastněných uživatelů o tom, že získali práva k novému projektu.

4.4 Technologie použitelné při vývoji webové aplikace

V této sekci popisují výhody a nevýhody knihoven a frameworků, které jsem zvolil pro vývoj tohoto projektu a důvody pro jejich použití.

4.4.1 Symfony

Symfony je webový aplikační framework a sbírka znovupoužitelných PHP¹knihoven a komponent. Jeho cílem je urychlit tvorbu a údržbu webových aplikací a omezit zbytečné a stále se opakující kusy kódu.

Framework Symfony je určen na tvorbu větších projektů a nabízí svým vývojářům plnou kontrolu nad nastavením takového projektu: od adresářové struktury až po použití knihoven třetích stran, přičemž téměř vše lze přizpůsobit potřebám navrhovaného systému. [4]

Projekty vytvořené v tomto frameworku jsou založené na architektuře MVC²a využívají výhody objektově orientovaných jazyků, čímž pomáhají vytvořit lehce testovatelný kód při jehož vývoji je výrazně šetřen čas vývojářů. Framework zároveň umožňuje vytváření a přidávání zásuvných modulů, které tak dávají aplikaci novou funkcionalitu bez nutnosti ji vyvíjet od nuly.

I přes delší učící křivku tohoto frameworku jsem se rozhodl využít jej jako základ pro implementaci tohoto projektu, a to jak pro jeho dobrou škálovatelnost, tak i pro dobrou podporu všech pro tento projekt potřebných komponent.

4.4.2 Bootstrap

Bootstrap je webový framework pro návrh front-endu webových aplikací a stránek. Skládá se z designových šablon založených na jazycích HTML³a CSS⁴, které slouží k popisu formulářů, tlačítek, navigačních prvků a jiných komponent uživatelského rozhraní. [19]

Velkou výhodou tohoto frameworku je jednoduchost jeho integrace do vytvářené aplikace, kdy lze jednoduše vylepšit vzhled téměř libovolného prvku pouze přidáním několika klíčových slov. Zároveň byl již při návrhu kladen velký důraz na výslednou responsivnost vytvářených stránek, které tak lze jednoduše přizpůsobit libovolné velikosti uživatelského displeje.

Tento framework jsem pro projekt zvolil pro jednoduchost jeho použití i pro mé předchozí pozitivní zkušenosti.

4.4.3 Doctrine

Projekt Doctrine je sbírka několika PHP knihoven určených pro ukládání dat do databáze a mapování objektů s těmito daty. Základem této sbírky jsou projekty Object Relational Mapper (ORM) a Database Abstraction Layer (DBAL), na jejichž základech je vystaven. [2]

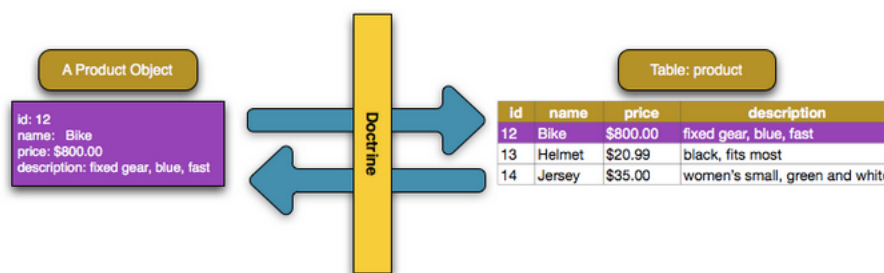
Doctrine využívá k trvalému ukládání dat třídy *Entit*, u kterých lze každý jejich atribut pomocí anotací namapovat na jeden sloupec relační databáze, jak je znázorněno na obrázku

¹PHP – *Hypertext Preprocessor* – je skriptovací programovací jazyk. <http://php.net/>

²MVC – *Model View Controller* – je softwarová architektura rozdělující datový model aplikace, uživatelské rozhraní a řídicí logiku do tří nezávislých komponent

³HTML – *HyperText Markup Language* – je značkovací jazyk používaný pro tvorbu webových stránek

⁴CSS – *Cascading Style Sheets* – je jazyk pro popis způsobu zobrazení elementů na webových stránkách



Obrázek 4.10: Schéma mapování jednoduchého objektu na databázovou tabulku pomocí nástroje Doctrine [4]

číslo 4.10. Samotné mapování je před vývojářem skryto tak, aby mohl jednoduše a rychle vytvářet nové jednoduché objekty. [4]

Důvodem výběru tohoto nástroje je především jeho přímá integrace ve frameworku Symfony výrazně zjednodušující trvalé ukládání dat v aplikaci.

4.4.4 Chart.js

Chart.js je knihovna založená na HTML5 a jazyku JavaScript vyznačující se hlavně svou jednoduchostí a snadností používání. S její pomocí lze vytvářet animované, interaktivní a lehce přizpůsobitelné grafy. [1]

Její výhodou je také podpora všech významných prohlížečů (včetně jejich starších verzí) a podpora responsivního návrhu.

Tuto knihovnu jsem si vybral zejména proto, že je open-source a je použitelná zdarma. Ačkoliv grafy v této knihovně nejsou na první pohled tak hezké, jako u některých jiných knihoven, podporuje mnohem širší modifikovatelnost, což je vzhledem k vlastnostem tohoto projektu a významu, jaký v něm hrají grafy, velice důležitý bod.

Kapitola 5

Implementace

Cílem tohoto projektu bylo navrhnout a implementovat webovou aplikaci se silnou uživatelskou interakcí na straně klienta. Takováto aplikace nutně potřebuje ukládat všechna uživatelská data do databáze a správně s těmito daty pracovat, ale také potřebuje jednoduché a snadno použitelné uživatelské rozhraní.

Proto je i tato kapitola rozdělena do dvou sekcí. V první sekci popisuji náročnější úkoly, se kterými jsem se potýkal na straně serveru, a způsob, jakým jsem se s nimi vypořádal. V druhé sekci jsou pak popsány nejdůležitější prvky grafického uživatelského rozhraní aplikace a způsob jejich implementace.

5.1 Serverová část

Všechna logika aplikace je napsána v programovacím jazyce PHP a je zasazena do frameworku Symfony.

5.1.1 Nasazení na webový server

Pro možnost aplikaci testovat a používat jinde než na lokálním serveru, na kterém byla vyvíjena, bylo nutné najít vhodný web hosting na kterém by bylo možné aplikaci nasadit. Po prozkoumání trhu a několika vhodných kandidátů jsem si nakonec vybral platformu Heroku¹, která je zdarma, nabízí množství rozšíření a přímo podporuje integraci frameworku Symfony. Nevýhodou této platformy je skutečnost, že aplikace, která není více než půlhodinu aktualizována libovolným uživatelem, je odstraněna z vyrovnávacích pamětí a její opětovné načtení trvá delší dobu (jednotky až desítky sekund), což by může některým uživatelům používání přinejmenším znepříjemnit.

Pro správné spuštění aplikace na této platformě bylo potřeba v konfiguraci nastavit některé proměnné prostředí Symfony jako jsou například `SYMFONY_ENV` a `APP_SECRET`.

Pro správnou funkci všech důležitých vlastností aplikace bylo také nutné nainstalovat v Heroku další rozšíření. Pro ukládání dat do databáze jsem využil rozšíření `ClearDB MySQL`². Pro rozesílání pozvánek z aplikace na e-mail jsem použil službu `SendGrid`³ nabízející až 100 odeslaných emailů denně zdarma. Pro kontrolu pravidelných, každodenních akcí, jako je například rozesílání pravidelných připomínek nebo upozornění uživatelů na blížící

se konec termínů, jsem využil rozšíření **Heroku Scheduler**⁴ které nabízejí podobné funkce jako například softwarový démon Cron.

5.1.2 Komunikace mezi klientem a serverem

Během vývoje aplikace byl kladen důraz na jednoduchost používání a uživatelskou přívětivost. Jedním ze způsobů, jak jsem se tohoto snažil dosáhnout, je odesílání a ukládání všech změn provedených uživatelem na server bez nutnosti znovu načítat a zobrazovat celou stránku.

Každá uživatelská změna je tak odesílána asynchronně, je na serveru zpracována a po obdržení kladné odpovědi je poté změna v aplikaci překreslena. Příkladem složitějších událostí může být například změna nastavení úkolu (změna barvy grafu, jeho názvu atd.), kdy je nutné ze serveru odeslat nové údaje o celém grafu, který je potřeba z dokumentu smazat, celý znovu vykreslit a znovu inicializovat všechny funkce reagující na pohyb myši v grafu (více v kapitole 5.2.1).

Nevýhodou tohoto přístupu je nutnost použití moderních webových prohlížečů se zapnutým jazykem JavaScript, což by ovšem v dnešní době neměl být žádný problém.

5.1.3 Autentizace uživatelů

Z důvodu požadavku na možnost okamžitě upravovat hodnotu úkolů bez nutnosti registrace plynoucí ze zadání práce je nutné provádět „registraci“ uživatelů automaticky. Každému nepřihlášenému uživateli, který si otevře aplikaci, je tak vždy vytvořen anonymní účet. Ten umožňuje sledovat všechny změny, které v aplikaci provede, a spojit je s tímto jedním (i když anonymním) uživatelem. Pokud si tento uživatel nezmění své jméno, nebo se nepřihlásí, nebude tak sice nikdo vědět, kdo přesně tyto změny provedl, ale bude alespoň možné říct, že všechny změny provedl jeden stejný člověk. Úpravy anonymními uživateli jsou navíc implicitně zakázané, takže administrátoři této vlastnosti nemusí využívat, pokud si nepřejí.

Pro implementaci tohoto požadavku bylo ovšem nutné změnit výchozí chování bezpečnostního modulu frameworku Symfony. Ten automaticky nepřihlášenému uživateli vytvoří anonymní *token*, který jej pak nadále provází aplikací. Toto chování bylo nutné potlačit a již při prvním načtení stránky uživatele přihlásit a udělit mu vlastní práva. Toho bylo dosaženo pomocí bezpečnostní komponenty **Guard**⁵ frameworku Symfony. Vytvořil jsem tak vlastní autentizátor rozšiřující třídu **AbstractGuardAuthenticator** a ten jsem zaregistroval v konfiguračním souboru `config/packages/security.yaml`. Tímto způsobem svěří bezpečnostní systém frameworku Symfony autentizaci uživatelů mé třídě a bude tedy možné vytvořit účet i anonymním uživatelům.

Systém si pamatuje identitu uživatele pomocí souborů *cookies* a při načtení každé stránky zkontroluje, zda již uživatel v databázi existuje. Potom zjistí práva uživatele k právě požadované stránce a buď povolí zobrazení stránky a její vykreslení, nebo uživatele přesměruje na stránku s chybovým hlášením.

¹<https://www.heroku.com/>

²<https://elements.heroku.com/addons/cleardb>

³<https://www.sendgrid.com/>

⁴<https://elements.heroku.com/addons/scheduler>

⁵https://symfony.com/doc/current/security/guard_authentication.html

5.1.4 Autentizace pomocí Google API

Pro přihlášení jsem se rozhodl využít autentizaci uživatelů pomocí společnosti Google. Díky tomuto se mohou uživatelé přihlásit pod svým již existujícím účtem od společnosti Google a nemusí se tak registrovat a pamatovat si nový účet pro jednu jedinou aplikaci. Naopak pro mou aplikaci tak odpadá starost s registračním formulářem, obnovováním zapomenutých hesel atd.

Pro implementaci bylo nejprve nutné aplikaci zaregistrovat u společnosti Google a zadat zde URL adresy ze kterých se budou uživatelé přihlašovat. Pro samotnou komunikaci se servery Googlu jsem použil knihovnu `HWIOAuthBundle`⁶ vytvořenou pro framework Symfony. Ta zajišťuje přesměrování na přihlašovací dialog, zpracuje odpověď od Googlu a vrátí mé aplikaci objekt s již načtenými údaji o uživateli.

V případě, že byl uživatel do aplikace pozván pomocí odkazu, získá uživatel okamžitě po zobrazení daného odkazu práva k danému objektu (více v následující podkapitole 5.1.5). Pokud to ovšem byla uživatelova první zkušenost s touto aplikací, získá tato práva jako anonymní uživatel (viz předchozí podkapitola 5.1.3). Když se ale v průběhu používání aplikace rozhodne přihlásit, všechna tato práva by byla ztracena. Proto jsou při každém přihlášení porovnány práva přihlášeného uživatele s jeho právy před přihlášením a v případě, že by vlastnil před přihlášením některá práva navíc, získá tato práva i pro svůj přihlášený účet.

Stejným způsobem proběhne kontrola, zda nebyl původní nepřihlášený uživatel přiřazen k některému z konkrétních úkolů (například pokud byl pozván e-mailem). V tomto případě ovšem může být u každého úkolu přiřazen pouze jeden uživatel, a tak je toto přiřazení odebráno nepřihlášenému uživateli a k jeho úkolům je přiřazen jeho Google účet.

K autentizaci není samozřejmě nutné využívat pouze služeb společnosti Google, jelikož knihovna `HWIOAuthBundle` poskytuje podporu pro více než 55 různých poskytovatelů. Pokud by tedy v budoucnu požadovala větší skupina uživatelů přidání dalších poskytovatelů, nebylo by jejich přidání do aplikace příliš složité. V rámci zachování jednoduchosti jsem se ovšem prozatím rozhodl nenabízet žádné další možnosti autentizace skrze účty jiné společnosti.

5.1.5 Sdílení objektů

Objektem se v rámci sdílení v aplikaci myslí buď celý projekt, nebo jen jedna z jeho záležitostí. Logika sdílení je ovšem podobná a pokud nebude uvedeno jinak, budou se informace zde uvedené vztahovat k oběma z těchto objektů.

Objekt lze v aplikaci sdílet třemi způsoby: hromadně pomocí odkazu, jednotlivě s uživatelem pomocí e-mailu, nebo přímo s uživatelem již přihlášeným v aplikaci.

V případě sdílení s uživatelem přímo je vše velmi přímočaré – jelikož již znám jeho identifikátor v databázi, jsou mu zvolená práva rovnou přidělena. Jinak je ovšem situace složitější, protože aplikace předem neví, který uživatel odkaz otevře a komu tedy daná práva přidělit.

Při pozvání uživatele přes e-mail je situace velmi podobná postupu popsanému níže pro sdílení skrze odkaz (protože e-mail také musí obsahovat odkaz do aplikace), jen s tím rozdílem, že se takto pozvaného uživatele nebudou týkat případné změny v nastavení práv sdíleného odkazu.

⁶<https://github.com/hwi/HWIOAuthBundle>

V případě sdílení pomocí odkazu lze pak pro odkaz nastavit tři různé typy práv, která může uživatel po zobrazení tohoto odkazu získat: právo položky v objektu pouze prohlížet bez možnosti cokoliv měnit, právo objekt upravovat, nebo právo objekt upravovat, ale pouze pokud se přihlásí pomocí svého Google účtu. Odkaz ke sdílení obsahuje URL objektu, ke kterému má uživatel získat práva, a poté dlouhý řetězec náhodných znaků zaručující, že nepůjde získat přístup k objektu například pomocí útoku silou.

Pokud uživatel ještě nevlastní žádná přístupová práva k objektu a zadal do prohlížeče správný řetězec (který mu buď přišel emailem, nebo jej s ním sdílel majitel projektu či kolega), obdrží v případě sdílení záležitosti práva podle následujícího postupu:

1. Přiděl uživateli práva k záležitosti na základě nastavení práv odkazu.
2. Přiděl uživateli práva k prohlížení celého projektu ke kterému záležitost náleží.
3. Pro každou ostatní záležitost v projektu přiděl uživateli práva k prohlížení.

I když tedy byla s uživatelem sdílena pouze jedna záležitost, získá práva k prohlížení celého projektu, ke kterému záležitost náleží. Toto chování umožňuje uživateli mít lepší přehled i o ostatních činnostech jeho spolupracovníků.

Práva náležející odkazu ke sdílení objektu lze kdykoliv změnit a v tomto případě jsou automaticky upravena i práva všech uživatelů, kteří získali přístup skrze daný odkaz, na nově zvolenou hodnotu.

V případě, že administrátor změní práva odkazu ke sdílení celého projektu, jsou aktualizovány i práva odkazu jeho jednotlivých záležitostí, což umožňuje jednodušší správu velkých projektů. To ovšem platí pouze v případě, že předtím nebyla práva odkazu záležitosti změněna zvlášť. Taková záležitost je potom „odpojena“ od nastavení práv celého projektu a nadále se s ním nesynchronizuje, a to až do okamžiku, kdy by byla tato práva změněna zpět na původní hodnotu celého projektu. V grafickém uživatelském rozhraní je pak toto spojení indikováno ikonkou řetězu, jak lze vidět na obrázku v podkapitole 5.2.4.

Sdílení pomocí odkazu lze také zcela zakázat. V tom případě odkaz přestane být aktivní a všichni uživatelé, kteří získali přístup přes daný odkaz, o tyto práva přijdou a daný objekt jim nadále nebude nikde zobrazován.

5.1.6 Uživatelská práva

V aplikaci jsou vytvořené dohromady čtyři různé úrovně uživatelských práv. Uživatel tak může mít u objektu práva k jeho plné administraci, práva k vytváření a některým méně destruktivním změnám, nebo pouze práva k prohlížení. U záležitosti pak ještě existuje právo k úpravě pouze jednoho úkolu a prohlížení všech ostatních úkolů.

V případě, že byl uživatel do aplikace pozván pomocí emailu nebo přímo skrze jeho uživatelské jméno v aplikaci, lze jeho práva měnit bez problémů. Ovšem v okamžiku, kdy uživatel získal přístup pomocí odkazu, a administrátor se rozhodne změnit pouze jeho práva (jinak než změnou práv celého odkazu), je uživatel od tohoto odkazu „odpojen“ a jakákoliv další změna v právech tohoto odkazu (viz předchozí podkapitola 5.1.5) se jej již nebude týkat až do doby, než administrátor případně změní jeho práva zpět na původní hodnotu odkazu, čímž uživatele s tímto odkazem opět „sváže“.

Pokud se administrátor rozhodne odebrat uživateli všechna práva, je potřeba zkontrolovat, jestli nebyl daný uživatel přiřazen k některému z úkolů v daném objektu. V tom případě bude uživatel od úkolu odebrán a ten zůstane bez přiřazeného uživatele. V opačném

případě by došlo k situaci, kdy je k úkolu přiřazen uživatel, který jej nemůže ani zobrazit, natož pak upravovat.

5.1.7 Náповěda

V rámci zpřístupnění aplikace co největšímu počtu uživatelů a také z důvodu seznámení nových uživatelů s některými méně obvyklými prvky (jako je například zaznamenávání změny úkolu pomocí táhnutí za sloupeček grafu) byl v aplikaci implementován systém nápověd pro nové uživatele.

Každému novému uživateli jsou v databázi vytvořeny řádky pro každou z nápověd indikující, že uživateli ještě nebyla zobrazena. Tyto řádky jsou společné pro anonymní i přihlášené uživatele a pokud se již nápověda zobrazila anonymnímu uživateli a ten se potom přihlásí, nebude mu znovu zobrazena.

Každá z nápověd obsahuje podmínku během které by se měla uživateli zobrazit (například po prvním kliknutí na určitý prvek, nebo po prvním vytvoření nějakého objektu), a pokud je tato podmínka splněna a uživateli daná nápověda ještě nebyla zobrazena, nápověda se uživateli zobrazí. Tu lze uzavřít kliknutím na libovolnou část obrazovky nebo zmáčknutím libovolné klávesy. Uživatel má také možnost nápovědy vypnout a v tom případě se mu již žádná další nezobrazí. Příklad dvou obrazovek s nápovědou je přiložen v příloze číslo C.3.

Seznam nápověd, které jsou v současné době v aplikaci implementovány:

- Náповěda po prvním přihlášení vysvětlující rozložení nástěnky uživatele a vybízející k vytvoření prvního projektu.
- Náповěda po vytvoření prvního projektu vysvětlující způsob rozložení a organizace projektu na menší celky zvané záležitosti (anglicky *issues*).
- Náповěda po vytvoření první záležitosti vysvětlující způsob reprezentace jednotlivých úkolů a jak lze změnit jejich nastavení.
- Náповěda vysvětlující způsob změny hodnoty úkolu pomocí táhnutí sloupečkem v grafu.
- Náповěda po změně hodnoty grafu vybízející uživatele k popsání vytvořené změny.
- Náповěda po zobrazení nastavení termínů vysvětlující jejich funkci.
- Náповěda po zobrazení nastavení pravidelných připomenutí vysvětlující jejich funkci.
- Náповěda vysvětlující způsob sdílení projektů a nastavení práv uživatelů.

5.1.8 Překlady

Všechny texty v aplikaci byly primárně psány v anglickém jazyce, ale mimo jiné i pro testování aplikace s uživateli ne tolik zdatnými v angličtině zde byla přidána možnost zobrazit aplikaci i v českém jazyce.

K překladu jednotlivých řetězců je využita vlastnost frameworku Symfony a jeho komponenty `Translations`⁷. Ta umožňuje abstrahovat jednotlivé texty do samostatné vrstvy a na základě proměnné prostředí vybrat odpovídající text ve zvoleném jazyce.

Většina textů v aplikaci je tak umístěna v samostatném `yml` souboru pro každý přeložený jazyk a framework samotný pak v čase překladu vybere správný řetězec. V současné době bylo do českého jazyka přeloženo pouze něco málo přes 100 vět či slov, což pokrývá pouze základní navigaci v aplikaci s tím, že zbytek textů zatím přeloženo není a jsou tedy v původním, anglickém jazyce.

Během budoucího vývoje by tedy bylo vhodné přeložit i zbytek řetězců. V případě zájmu o aplikaci mezi jinak mluvícími uživateli je však již celý systém překladů připraven a přidávání nových jazyků by bylo velmi jednoduché.

5.1.9 Databáze aplikace

Databáze je v aplikaci vytvářena automaticky pomocí knihovny `Doctrine ORM`⁸, která je integrována v systému Symfony. Místo vytváření tabulek a psaní SQL dotazů jsou jednotlivé tabulky reprezentovány PHP objekty (zvanými *entity*), které tato knihovna sama převede do SQL databáze. Tento způsob značně usnadňuje objektový návrh aplikace i psaní složitějších databázových dotazů.

Na obrázku číslo 5.1 je zobrazen diagram všech entit vytvořených pro tuto aplikaci, včetně jejich atributů a vztahů mezi nimi. V následujícím textu je pak velmi stručně vysvětlen význam každé z entit v aplikaci. Některé entity (především pro projekt a záležitost) mají velké množství společných atributů, které byly vyřešeny pomocí abstraktních tříd, od kterých tyto entity společné atributy dědí.

Board – objekt reprezentující jeden projekt. Dědí od objektu `AbstractEntity`.

BoardHistory – slouží k zaznamenání události kdy uživatel získá práva k projektu skrze odkaz ke sdílení. Dědí od objektu `AbstractHistoryEntity`.

BoardShare – uchovává uživatelská práva ke konkrétnímu projektu. Dědí od objektu `AbstractRoleEntity`.

Issue – objekt reprezentující jednu záležitost. Dědí od objektu `AbstractEntity`.

IssueHistory – zaznamenává získání práv k záležitosti skrz odkaz ke sdílení. Dědí od objektu `AbstractHistoryEntity`.

IssueShare – uchovává uživatelská práva k záležitosti. Dědí od objektu `AbstractRoleEntity`.

Gauge – objekt reprezentující jeden úkol.

GaugeRights – zaznamenává uživatelská práva k úkolu.

GaugeChanges – zaznamenává každou změnu stavu úkolu.

⁷<https://symfony.com/doc/current/translation.html>

⁸<https://symfony.com/doc/current/doctrine.html>

User – objekt reprezentující uživatele.

UserShare – slouží k zaznamenání pozvánky jednotlivého uživatele do aplikace.

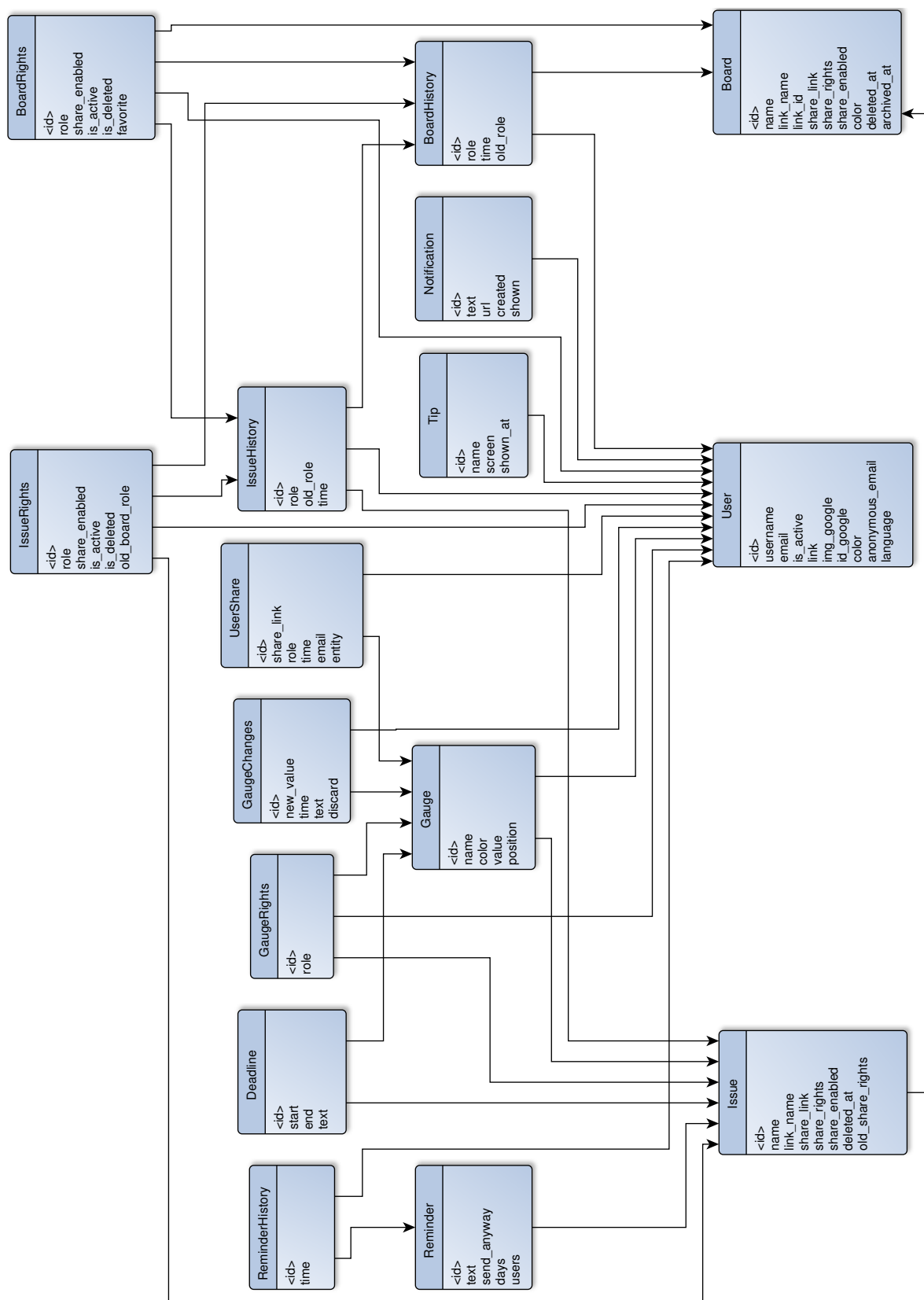
Reminder – zaznamenává nastavení pravidelných připomínek záležitosti.

ReminderHistory – slouží k zaznamenání události rozeslání pravidelné připomínky uživateli.

Tip – zaznamenává zobrazení jednotlivých nápověd uživatelem.

Deadline – objekt reprezentující jeden termín.

Notification – slouží k uchování připomínek zaslaných uživateli.



Obrázek 5.1: Diagram všech objektů v databázi aplikace

5.2 Grafické uživatelské rozhraní

Generování jednotlivých HTML stránek je zajištěno pomocí nástroje Twig⁹, který je součástí projektu Symfony. Jedná se o moderní šablonovací nástroj optimalizovaný pro jazyk PHP. Pomocí tohoto nástroje lze oddělit aplikační vrstvu od prezentační a nejsou tak dohromady míchány jazyky HTML a PHP.

Následující podkapitoly popisují nejdůležitější prvky grafického uživatelského rozhraní aplikace a způsob jejich implementace.

5.2.1 Zobrazení grafu s úkoly

Pro zobrazení grafu záležitosti s jednotlivými úkoly byla použita knihovna `Chart.js`¹⁰. Ta umožňuje jednoduchým způsobem zobrazovat na pohled líbivé grafy a zároveň poskytuje široké možnosti pro vlastní úpravy pomocí zásuvných modulů.

Pro zobrazení grafu bez možnosti jeho úpravy (například na nástěnce projektu nebo při zobrazení uživateli s právy pouze ke čtení) je tato knihovna upravena pouze minimálně. Jedinou změnou je registrace jednoho zásuvného modulu, který nad každým sloupečkem grafu zobrazí název daného úkolu. V případě, že je hodnota sloupce menší než 75 %, je název zobrazen nad daným sloupcem. V opačném případě už ale hrozí, že by tento text zasahoval mimo okno grafu, a je tedy vykreslen přímo přes sloupec. Vzhledem k tomu, že je tento text tmavý a sloupce grafu mohou mít také tmavou barvu je za tento text ještě přidán bílý stín pro lepší čitelnost.

Zároveň, pokud je název grafu delší než 14 znaků (toto číslo bylo získáno experimenty s různým množstvím sloupců i velikostí grafu) a název úkolu se skládá alespoň ze dvou slov, je tento text rozdělen na dva řádky zobrazené pod sebou tak, aby text nezasahoval do vedlejších sloupců.

V případě, kdy je potřeba změnit hodnotu úkolu táhnutím za sloupec grafu, bylo ovšem nutné tuto knihovnu rozšířit mnohem mohutněji, protože tuto funkci tato, ani žádná mně známá knihovna nenabízí. Bylo tedy vytvořeno rozšíření této knihovny napsané v jazyce JavaScript sloužící k tomuto účelu.

Toto rozšíření zaznamenává pohyby myši nad plochou s grafem a čeká na událost stisknutí levého tlačítka myši. V případě stisku porovná souřadnice této události s boxy jednotlivých sloupečků grafu a pokud je nalezen průnik, znamená to, že uživatel stiskl jeden z úkolů a chtěl by změnit jeho hodnotu.

V tom případě skript načte práva uživatele k jednotlivým sloupečkům a zjistí, zda uživatel může měnit hodnotu stisknuté sloupce. Pokud může, přeje skript do „aktivního režimu“, ve kterém zůstane do doby, dokud uživatel drží levé tlačítko myši stisknuté a nespustí se událost `onmouseup`. V tomto aktivním režimu skript registruje každý pohyb myši a porovnává jeho ypsilonovou hodnotu s umístěním plochy grafu v rámci okna prohlížeče. Pokud se myš pohybuje v rámci plochy grafu, je s každým jejím pohybem graf překreslen s novou hodnotou aktivního úkolu v závislosti na aktuální poloze myši. V případě, že je myš již nad grafem, je aktivní režim ukončen a hodnota sloupce je nastavena na 100 %. Stejný případ nastane i pokud se myš nachází pod plochou grafu, jen je sloupec nastaven na hodnotu 0 % (respektive 2 %, viz níže v této podkapitole).

V případě, že je tento aktivní režim ukončen přímo skriptem, nebo uživatel pustí levé tlačítko myši, odešle se aktuální hodnota grafu na server a skript nadále přestane reagovat na

⁹<https://twig.symfony.com/>

¹⁰<http://www.chartjs.org/>

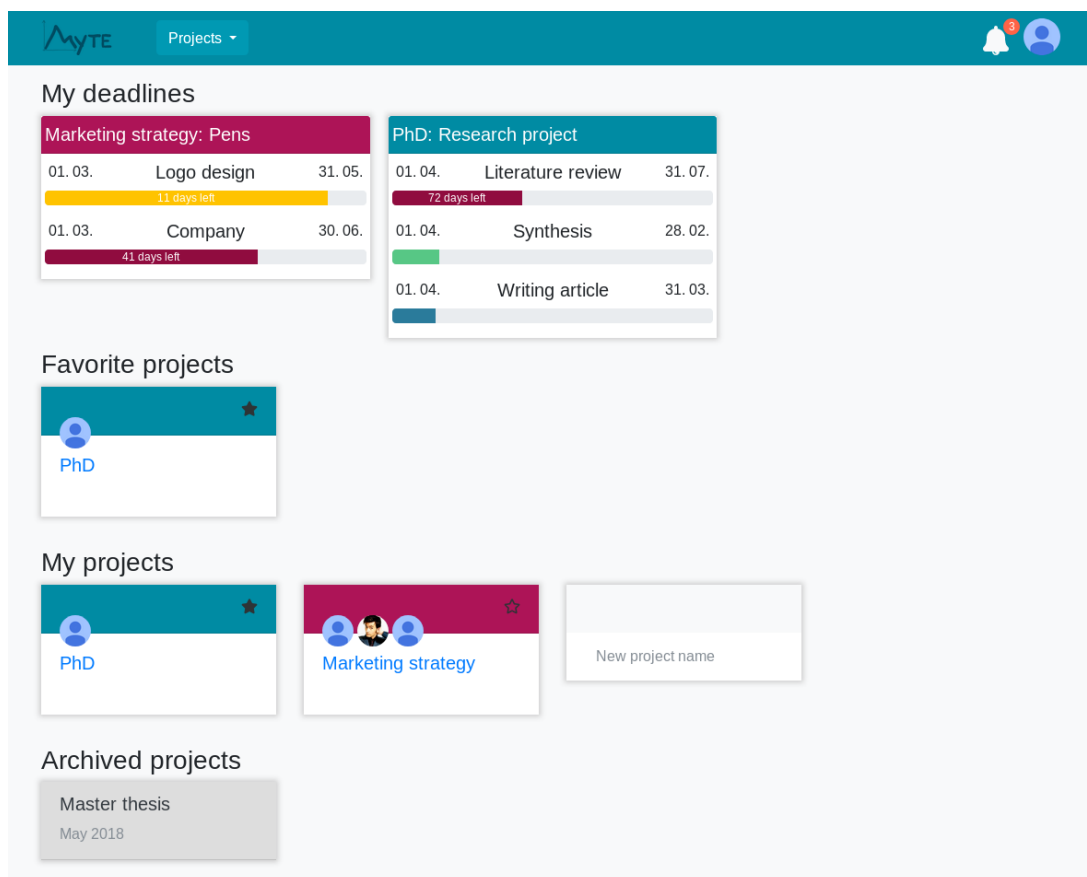
pohyb myši. Naopak, dokud uživatel právě provedenou změnu nepotvrdí (spolu s případným popisem provedené změny), nebo ji nezruší, skript ignoruje všechny další události myši.

Každý nový úkol má nastavenou výchozí hodnotu 0 %, což ovšem velice znesnadňuje uchopení tohoto grafu a jeho změnu nad výchozí hodnotu. Proto byla jako nejnižší možná zobrazitelná hodnota v grafu zvolena hodnota 2 %, aby byla vždy zobrazena alespoň část sloupce a uživatel jej měl za co „uchopit“. Jakákoliv nižší hodnota úkolu je tedy ignorována a převedena na hodnotu 2.

Na dotykových zařízeních se ovšem JavaScript chová odlišně a místo události `onmousedown` je tak nutné sledovat také událost `ontouchstart` a podobně i pro pohyb myši/prstu a jeho ukončení. Navíc jsou souřadnice dotyku na obrazovce odlišné od souřadnic získaných při stisku tlačítka myši a je proto nutné provést přepočty pro zjištění, zda se uživatel dotknul některého z úkolů v grafu. Po tomto přepočtu však již lze postupovat stejně jako při události stisku myši a lze využít již jednou implementované funkce.

5.2.2 Nástěnka uživatele

Nástěnka uživatele byla navržena v podkapitole 4.2.1 a jejím hlavním účelem je poskytnout přihlášenému uživateli přehled aktuálních termínů i rozcestník mezi jeho projekty.



Obrázek 5.2: Úvodní obrazovka přihlášeného uživatele obsahující seznam jeho úkolů a seznam projektů kterých je součástí.

Na obrázku číslo 5.2 je snímek z finální verze aplikace, který uživatel uvidí ihned po vstupu na stránku aplikace. V první řadě je zde zobrazen přehled všech uživatelových termínů navržený v podkapitole 4.3.1. Termíny jsou seskupeny dohromady podle jednotlivých záležitostí a obarveny barvou daného projektu, ke kterému náleží. Každá tato skupina pak obsahuje prvky **progress bar** v barvách daného úkolu reprezentující dobu uplynulou od začátku termínu.

V další části nástěnky je seznam všech projektů, ke kterým má uživatel přístupová práva. Každý projekt je zde reprezentován svým názvem, barvou a jsou zde zobrazeni všichni členové spolupracující na daném projektu. V případě, že na projektu spolupracuje více než pět uživatelů, jsou zde zobrazeni pouze čtyři neaktivnější, a dále se zobrazí ikonka s počtem zbývajících členů. Po kliknutí na tuto ikonku se pak uživateli zobrazí modální okno s výpisem všech uživatelů spolupracujících na projektu. Příklad tohoto okna je umístěn v příloze C.5.

Kromě toho je v rohu karty projektu umístěna ikonka hvězdičky umožňující uživatel přidat daný projekt mezi své oblíbené. Oblíbené projekty se mu zobrazují na prvních pozicích a jsou také jako první zobrazeny v rychlém menu ve vrchní navigační liště.

Jako poslední karta je uživateli nabídnuta možnost rychle vytvořit nový projekt. Po kliknutí na tuto kartu dostane uživatel možnost vybrat pro projekt jednu z přednastavených barev a zadat název nového projektu, jak je zobrazeno na obrázku v příloze C.6. Tvorba nového projektu je ovšem povolena pouze přihlášeným uživatelům a v případě, že uživatel již je součástí některého projektu, ale jedná se o nepřihlášeného, anonymního uživatele, (viz podkapitola Uživatelská práva na straně 32) je místo této karty zobrazena informace o tom, že pro tvorbu nového projektu je nutné, aby se uživatel přihlásil a je zde přidán i odkaz k přihlášení.

V poslední části této nástěnky je poté umístěn seznam projektů, ke kterým má uživatel přístupová práva, ale jsou již archivovány a uživatel má tedy možnost si již pouze prohlížet jejich hodnoty. Tyto projekty jsou zobrazeny tak, aby byly co nejméně výrazné a uživatele zbytečně nerozptylovaly.

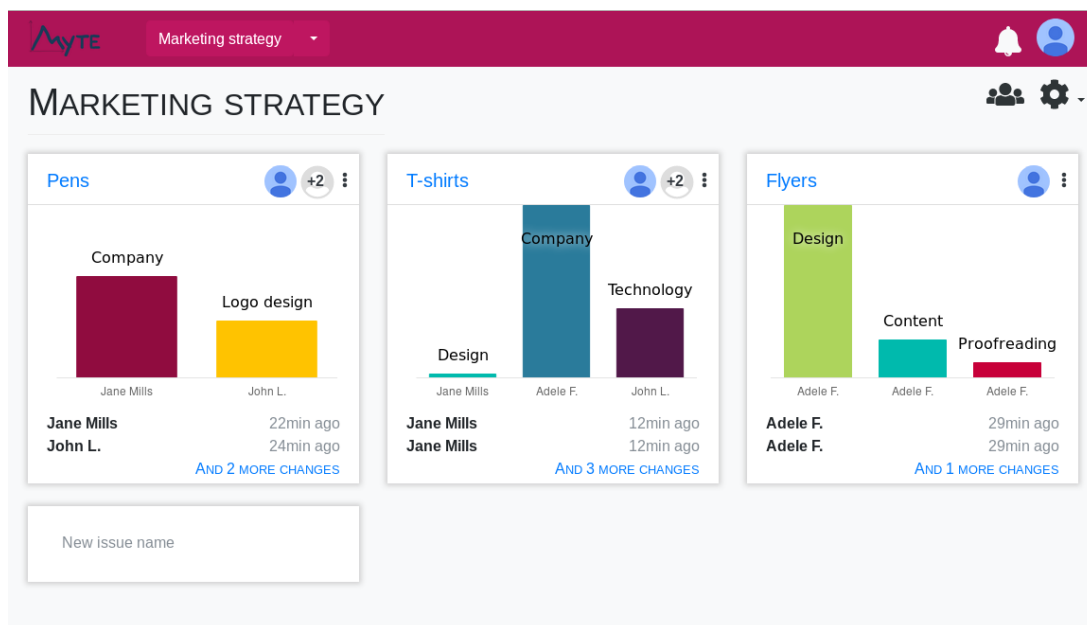
5.2.3 Nástěnka projektu

Nástěnka projektu byla navržena v podkapitole 4.2.2 a obsahuje menu s nastavením celého projektu a také seznam karet se všemi záležitostmi náležícími k danému projektu.

Na obrázku číslo 5.3 je uveden příklad této obrazovky pro projekt s názvem „marketing strategy“. V tomto konkrétním příkladu zvolil správce při vytváření projektu červenou barvu, která obarvila celou vrchní navigační lištu a pomáhá tak uživateli lépe se v aplikaci zorientovat (hlavně pokud pracuje na více projektech současně).

V pravé části stránky s nástěnkou je umístěna nabídka s možnostmi tento projekt spravovat. Ta se skládá ze dvou různých ikon: ikonou pro správu uživatelů a ikonou pro nastavení projektu. Pod správou uživatelů se administrátorovi projektu zobrazí možnost projekt sdílet (popsaná v podkapitole 5.2.4) a spravovat všechny uživatele projektu (popsaná v podkapitole 5.2.5). Uživateli, který nemá administrátorská práva, ale získal práva k projektu přes sdílený odkaz se místo této nabídky zobrazí možnost prohlížet si daný odkaz. Jinak se mu tato nabídka nezobrazí vůbec.

Po kliknutí na druhou ikonu s ozubeným kolem se administrátorovi projektu zobrazí možnost změnit název a barvu projektu (příklad tohoto okna je na obrázku v příloze C.8), možnost projekt smazat, archivovat nebo duplikovat (duplikace je popsána v podkapitole



Obrázek 5.3: Obrazovka konkrétního projektu se seznamem všech jeho záležitostí.

4.3.2). Ostatním uživatelům bez administrátorských práv k projektu se pak zobrazí pouze možnost projekt opustit nebo duplikovat.

Pod názvem projektu je umístěn seznam s kartami jednotlivých záležitostí. V horní části karty je zobrazen název záležitosti a seznam aktivních členů. Ten po rozkliknutí zobrazí seznam všech uživatelů záležitosti, který je podobný seznamu popsaném v předchozí podkapitole Nástěnka uživatele a jehož příklad je zobrazen na obrázku v příloze číslo C.5. Uživatel s právy správce k dané záležitosti se zde navíc zobrazí nabídka s několika rychlými akcemi týkající se záležitosti: odkaz k jejímu sdílení, možnost spravovat její uživatele a možnost smazat danou záležitost.

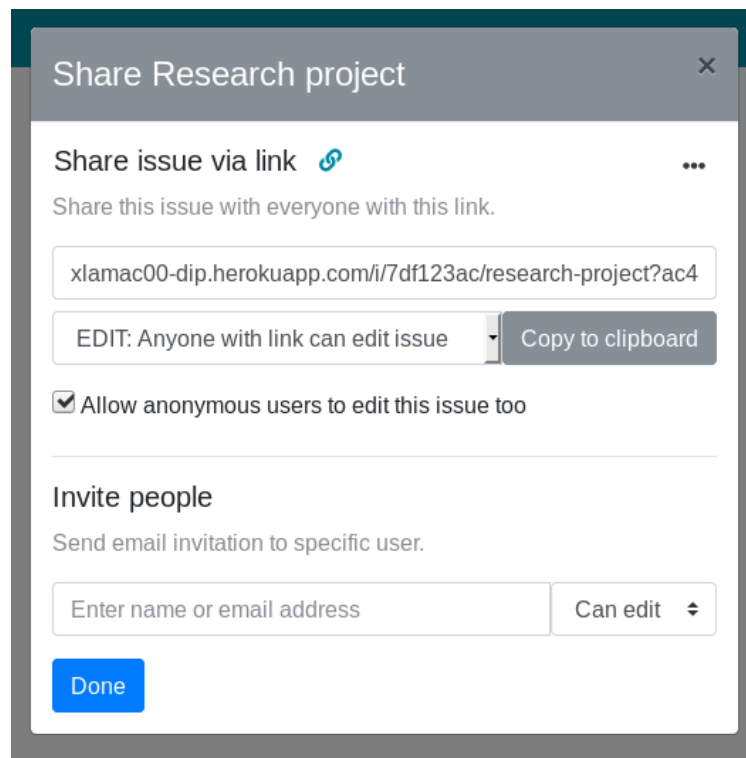
V prostřední části karty je zobrazen graf se stavem jednotlivých úkolů a pod ním jsou vypsány dvě poslední změny v záležitosti, včetně jména uživatele, který změnu provedl, a datum této změny. Původně zde byl zobrazen i název úkolu, u kterého ke změně došlo, včetně jeho nové a staré hodnoty, ale tato informace se nyní zobrazuje pouze po najetí na řádek s danou změnou, protože tento prvek při testování působil na uživatele příliš rušivě.

5.2.4 Sdílení objektů

Při kliknutí na ikonku sdílení (ať již projektu nebo záležitosti) se uživateli zobrazí modální okno s možností daný projekt sdílet.

Na obrázku číslo 5.4 je zobrazen příklad okna pro sdílení záležitosti. Okno pro sdílení projektu vypadá téměř stejně, jen je pozadí jeho hlavičky vybarvené barvou projektu místo šedé pro snadnou orientaci uživatele v těchto modálních oknech (ze stránky s nástěnkou projektu lze zobrazit obě tato modální okna).

Modální okno pro sdílení je rozděleno na dvě části: sdílení skrze odkaz a zvaní jednotlivých uživatelů. V první části je zobrazen vygenerovaný odkaz pro sdílení objektu a administrátor zde má možnost nastavit práva vztahující se k tomuto odkazu (způsob implementace těchto práv je popsán v podkapitole 5.1.5). Nad tímto odkazem je vrchní řádek obsahující



Obrázek 5.4: Modální okno s nastavením sdílení záležitosti a možností přizvat jednotlivé uživatele.

rozklikávací nabídku s možností sdílení přes odkaz naprosto zakázat, a v případě modálního okna pro sdílení záležitosti i ikonku s celým nebo přerušným řetězem symbolizující, zda je nastavení sdílení záležitosti svázáno s nastavením sdílení celého projektu, nebo bylo změněno a změny v nastavení sdílení projektu se jej netýkají (více v kapitole 5.1.5).

V druhé části okna má pak administrátor možnost pozvat jednotlivé uživatele. Pomocí knihovny `Bootstrap typeahead`¹¹ jsou zde uživateli po zadání několika prvních znaků do textového pole nabízeni možní uživatelé této aplikace, kteří již s daným uživatelem někdy spolupracovali. Uživatel zde pak má také možnost zadat e-mailovou adresu uživatele, kterého si přeje k projektu přizvat. V obou případech pak zadaný uživatel obdrží pozvánku k projektu a získá k němu přístupová práva dle volby administrátora.

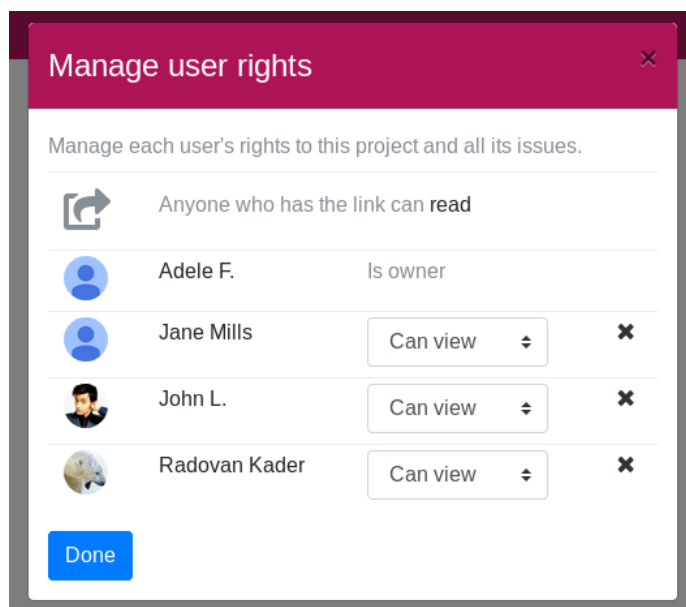
V případě, že uživatel nemá administrátorská práva k danému objektu, zobrazí se mu pouze samotný odkaz ke sdílení a textová informace o tom, jaká práva jsou k tomuto odkazu nastavena. Všechno ostatní nastavení, včetně možnosti zvát jednotlivé uživatele, je pak uživateli skryto.

5.2.5 Správa uživatelů

V případě, že uživatel v nastavení klikne na možnost „správa uživatelů“, zobrazí se mu modální okno se seznamem uživatelů, kteří mají k danému objektu přístupová práva (tento dialog je pro správu uživatelů projektu a záležitosti téměř stejný). V tomto okně pak lze uživatelům jejich práva změnit a odebírat.

¹¹<https://twitter.github.io/typeahead.js/>

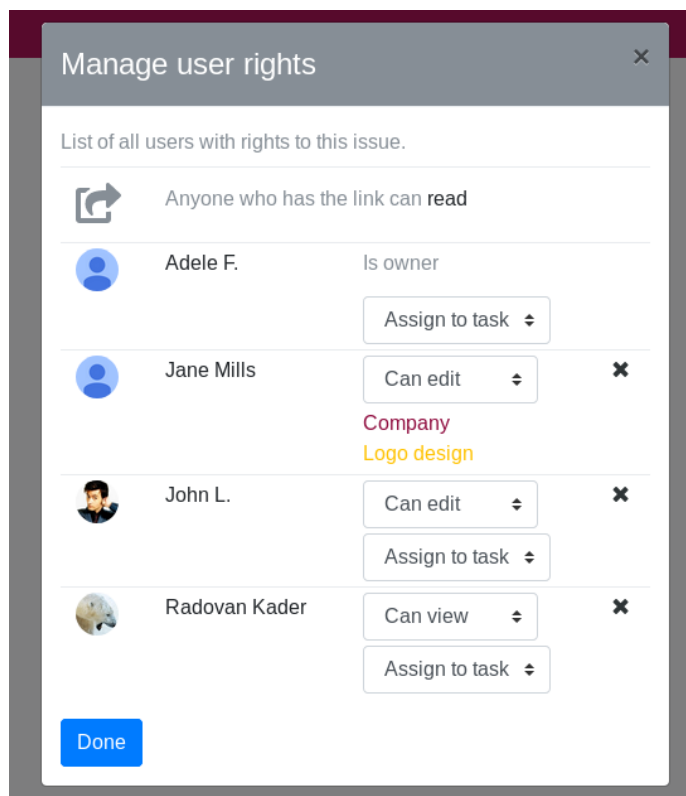
Na obrázcích 5.5 a 5.6 je příklad modálních oken pro správu uživatelských práv projektu a záležitosti. Tato dvě okna se v zásadě liší pouze barvou pozadí hlavičky (u projektu sdílí barvu celého projektu, u záležitosti je šedá) a v případě záležitosti je zde možnost přiřadit jednotlivé uživatele ke konkrétnímu úkolu.



Obrázek 5.5: Modální okno pro správu uživatelských práv projektu.

V obou případech jsou zobrazeni následující uživatelé: administrátoři daného objektu, uživatelé, kteří byli k objektu přímo pozváni (více možnost „pozvat jednotlivé uživatele“ v předchozí podkapitole 5.2.4) a přihlášení uživatelé, kteří k objektu získali přístupová práva skrze odkaz. Nejsou zde zobrazeni anonymní uživatelé, kteří práva získali přes odkaz. Prvním důvodem pro toto chování je, že by tento seznam mohl být u větších projektů neúnosně dlouhý a nepřehledný a za druhé by stejně správa těchto uživatelů příliš nedávala smysl: jejich přístupová práva lze měnit skrze práva odkazu ke sdílení a administrátor by pravděpodobně nevěděl, kterému anonymnímu uživateli právě práva mění. Toto chování je zároveň běžné i u velkých komerčních aplikací umožňujících sdílení anonymním uživatelům (jako jsou například cloudové služby společnosti Google, aplikace Dropbox a další).

Oproti oknu s právy uživatelů projektu (na obrázku 5.5) je u okna s právy k záležitosti (na obrázku 5.6) u každého uživatele jeden řádek navíc: ten obsahuje rozbalovací menu se jmény všech úkolů u záležitosti a umožňuje přiřadit uživatele ke konkrétnímu úkolu (více o této funkci je popsáno v podkapitole 6.2.3). U uživatele, který už je k některým úkolům přiřazen, je navíc zobrazen seznam všech jeho úkolů (s názvem v barvě sloupečku daného úkolu), pomocí kterého lze uživateli úkol opět odebrat.



Obrázek 5.6: Modální okno pro správu uživatelských práv záležitosti včetně možnosti přiřadit uživatele ke konkrétnímu úkolu.

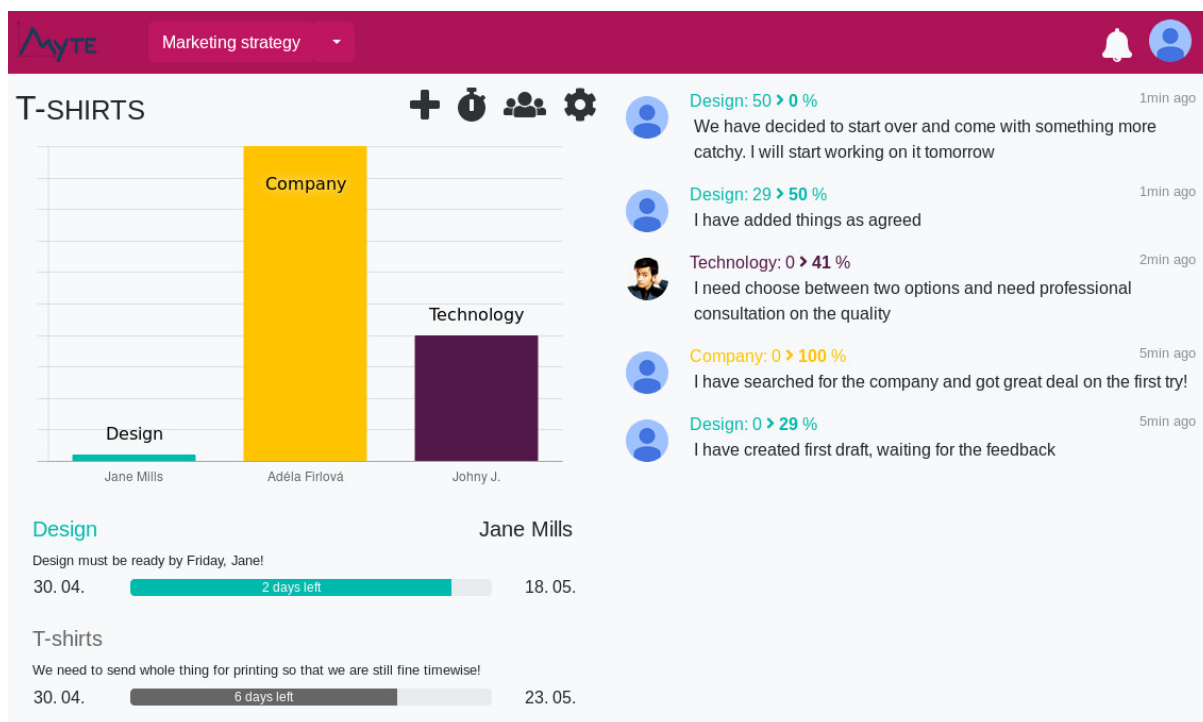
5.2.6 Detail záležitosti

Po kliknutí na název záležitosti se zobrazí stránka s jejím detailem navržená v podkapitole 4.2.3. Ta obsahuje graf se stavem jednotlivých úkolů, kterým lze po kliknutí a tažení libovolným úkolem měnit jeho hodnotu. Dále obsahuje seznam všech změn provedených v záležitosti seřazených od nejnovější k nejstarší a seznam všech termínů vztahujících se k dané záležitosti. V neposlední řadě lze na této stránce upravovat celou záležitost a měnit její nastavení.

Na obrázku číslo 5.7 je zobrazen příklad této stránky. Ta je dle původního návrhu rozdělena na dvě poloviny: v první polovině je název záležitosti a řádek s tlačítky pro správu záležitosti, graf s jednotlivými úkoly a seznam všech termínů. V druhé polovině jsou pak zobrazeny všechny změny provedené uživateli včetně jejich komentáře, data provedení změny, názvu úkolu a jeho staré a nové hodnoty.

První z tlačítek umožňuje přidat k záležitosti nový úkol. Toto tlačítko je pouze aktivní pokud má uživatel práva provádět v záležitosti změny a pokud existují pouze tři či méně úkolů. V případě, že uživatel získal práva k záležitosti přes odkaz, ale u odkazu nebyla povolena možnost úpravy anonymním uživatelům a aktuální uživatel není přihlášený, je uživatel informován o tom, že aby mohl záležitost upravovat, je nutné se přihlásit.

Druhé z tlačítek umožňuje vytvářet a upravovat termíny. Další tlačítko umožňuje měnit uživatelská práva a téměř se neliší od stejně vypadajícího tlačítka popsaného v podkapitole číslo 5.2.3. Poslední tlačítko s ikonou ozubeného kola zobrazí uživateli s administrátorskými



Obrázek 5.7: Stránka s detailem záležitosti zobrazující graf s hodnotami jednotlivých úkolů, seznam termínů a seznam všech změn úkolů.

právy následující možnosti: upravovat jednotlivé úkoly (popsané v následující podkapitole 5.2.7), změnit jméno záležitosti a nastavit pravidelné připomínky (popsané v podkapitole 5.2.8).

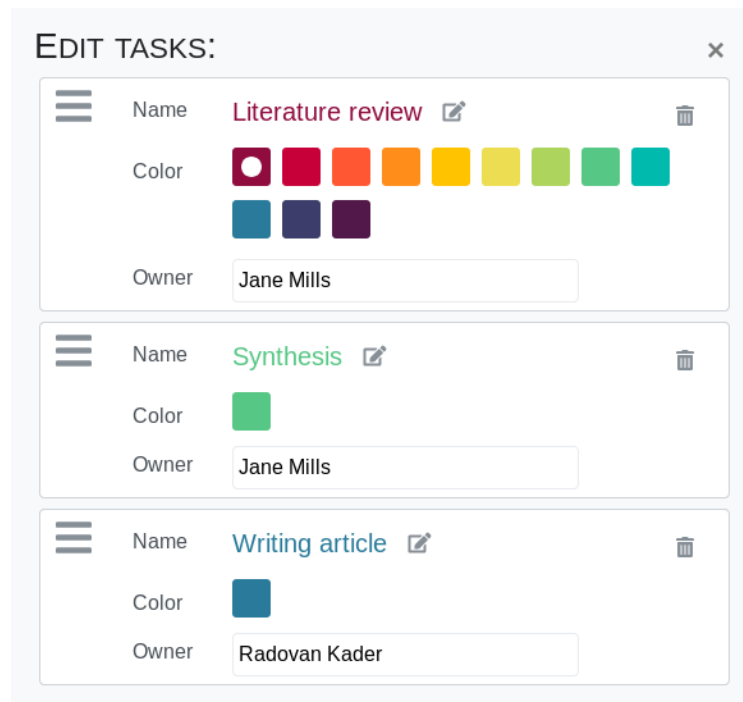
5.2.7 Úprava záležitosti

Po kliknutí na tlačítko úpravy jednotlivých úkolů na stránce s detailem záležitosti je obsah pravé poloviny této stránky nahrazen oknem navrženým v podkapitole 4.2.4.

Příklad tohoto okna je zobrazen na obrázku číslo 5.8. Dané okno obsahuje seznam všech úkolů v záležitosti a u každého úkolu umožňuje uživatelům s právy k úpravě záležitosti měnit jméno a barvu úkolu, případně daný úkol odstranit. Po kliknutí na políčko s barvou úkolu se uživateli zobrazí možnost vybrat jednu z dvanácti přednastavených barev, jak lze vidět na obrázku 5.8 u prvního úkolu.

Administrátor na této stránce zároveň může k úkolům přiřadit konkrétního uživatele. K tomuto slouží textové pole, do kterého lze zadat buď emailovou adresu nového uživatele, který posléze obdrží pozvánku do aplikace s právy k úpravě pouze daného úkolu, nebo zde lze vybírat pomocí jména z uživatelů, kteří již mají v aplikaci vytvořený účet. K implementaci tohoto výběru jsem využil funkci našeptávání knihovny **Bootstrap typeahead**.

Pomocí ikonky v levé části karty s úkolem je také možné měnit pořadí jednotlivých úkolů v grafu. Tato funkce je implementována pomocí funkcí **HTML5 Drag and drop**.



Obrázek 5.8: Stránka s možností editace úkolů v záležitosti.

5.2.8 Pravidelná připomenutí

Jednou z funkcí, která vyplynula z požadavků zadavatele, je možnost posílat všem zúčastněným členům pravidelná připomenutí. Tuto funkci lze využít například v případě, kdy tým spolupracující na projektu pořádá pravidelné týdenní schůzky, případně pokud by vedoucí daného týmu chtěl zaručit, že jeho uživatelé v aplikaci minimálně jednou týdně aktualizují stav svého postupu.


Příklad implementace uživatelského rozhraní pro nastavení této funkce lze vidět na obrázku číslo 5.9. V první části rozhraní je zobrazen seznam ikon všech uživatelů, kteří mají u projektu právo měnit alespoň jeden úkol. Z tohoto seznamu může administrátor kliknutím na ikonu odebrat libovolné členy, u kterých si nepřeje, aby tato pravidelná připomenutí dostávali (může se jednat například o ostatní správce, kteří mají díky své roli také práva k úpravě záležitosti, ale jinak v ní slouží pouze jako pozorovatelé a aktivně se do záležitosti nezapojují). To, že uživatel upozornění dostávat nebude, je pak indikováno zešednutím jeho ikonky. Tento údaj je pak v databázi uchován jako seznam „vynechaných“ uživatelů, raději než seznam uživatelů, kterým má upozornění chodit. To je důležité především v případě, kdy do záležitosti přibude nový uživatel, který je tímto bez jakéhokoliv dalšího zásahu do databáze nebo nastavení záležitosti také automaticky přidán do seznamu pravidelných připomínek.

Pod výběrem uživatelů je možnost vybrat si konkrétní dny v týdnu, kdy se mají tyto připomínky rozesílat. Nelze zde zvolit interval větší než jeden týden, jelikož zaznamenání například pouze jedné změny měsíčně je dle mého názoru známkou špatného projektového řízení a tato možnost by tedy zbytečně komplikovala toto nastavení.

V neposlední řadě zde může uživatel vybrat, zda si přeje, aby chodilo připomenutí i uživatelům, kteří již od posledního připomenutí zaznamenali nějakou další změnu, nebo aby

WEEKLY PROGRESS: x

Every selected day in the morning will send notifications to all users contributing to this issue to remind them to update their progress.



Mo Tu We Th Fr Sa Su

☐ Send notifications even if user already made any new change since last notification.

[Change email text](#) ▾

[Save changes](#)

Obrázek 5.9: Týdenní připomenutí uživatelům, aby aktualizovali svůj postup na projektu s možností vybrat si dny odeslání připomenutí

byli informováni pouze ti uživatelé, kteří nic nového nezaznamenali a na tento záznam tak například zapomněli. Využití informování i aktivních uživatelů je vhodné také v případě, že tým pořádá pravidelné týdenní schůzky a vedoucí projektu si přeje, aby se všichni uživatelé na danou schůzi připravili a zaznamenali své nejnovější změny.

Funkce, která je spuštěna každý den ráno a má na starost odesílání těchto připomenutí tak funguje přibližně takto:

- Vyber z databáze všechny záležitosti, které mají v nastavení připomínek zaškrtnutý aktuální den v týdnu a pro každou záležitost proveď:
 - Vyber z databáze všechny uživatele, kteří mají u záležitosti právo k zápisu a pro každého z nich proveď:
 1. Zkontroluj, zda od poslední připomínky v aplikaci provedl nějakou změnu, případně zda je v nastavení připomínky zvoleno informování všech uživatelů. Pokud od poslední připomínky nějakou změnu zaznamenal a nemají být informováni všichni uživatelé, uživatele přeskoč. Jinak pokračuj.
 2. Zkontroluj, zda uživatel není na seznamu vynechaných uživatelů, kterým připomínky chodit nemají. Pokud zde je, uživatele přeskoč. Jinak pokračuj.
 3. Zkontroluj, zda se jedná o přihlášeného uživatele, případně zda je u něj v databázi uložená emailová adresa. Pokud je, zašli mu email s připomínkou.
 4. Vytvoř v databázi novou notifikaci pro uživatele vyzývající k aktualizování stavu úkolu.

V případě nasazení aplikace na vlastní doméně se dá pro spouštění této funkce využít softwarový démon **Cron**. Při nasazení aplikace na platformě **Heroku** bylo využito rozšíření **Heroku Scheduler**.

5.2.9 Navigační panel

Navigační panel slouží v aplikaci k rychlé orientaci uživatele a obsahuje tři základní prvky: tlačítko s názvem aktuálního projektu a nabídkou ostatních projektů, seznam uživatelových notifikací (popsaný v následující podkapitole 5.2.10) a možnost spravovat nastavení uživatele. Barva tohoto panelu je také změněna v závislosti na barvě projektu, který si uživatel zrovna prohlíží.

Hlavním důvodem pro tlačítko s názvem aktuálního projektu je jednoduchá orientace uživatele v aplikaci. Uživatel tak vždy ví, který projekt si zrovna prohlíží a po kliknutí na toto tlačítko je přesměrován na stránku s nástěnkou daného projektu. Vedle tohoto tlačítka je také ikonka, která slouží k zobrazení nabídky všech uživatelových projektů a nabízí tak možnost rychlého přepínání mezi jednotlivými projekty. Pod touto nabídkou je pak také možnost vytvořit projekt nový. Příklad zobrazení této nabídky je na obrázku v příloze číslo C.2.

Druhým důležitým prvkem je nabídka s nastavením uživatele. Zde lze změnit uživatelské jméno v aplikaci (které je jinak buď „Anonymous user“, nebo je převzato z nastavení jeho Google účtu) a jazyk ve kterém budou texty v aplikaci napsány. Anonymní uživatelé zde mají možnost přihlásit se do aplikace pomocí Google účtu, zatímco přihlášení uživatelé mají naopak možnost odhlásit se. Příklad zobrazení této nabídky je uveden v příloze na obrázku číslo C.3.

Během budoucího vývoje aplikace by bylo vhodné tuto nabídku upravit a nabídnout uživateli samostatnou obrazovku s širší možností nastavení svého účtu. Mezi prvky, které by bylo vhodné uživateli nabídnout je například možnost změnit uživatelský obrázek (který je momentálně stažena z jeho účtu u společnosti Google a již jej nelze v aplikaci změnit), možnost upravit si nastavení zasílaných notifikací a emailů a jiné.

Dále by bylo vhodné na tento panel přidat také možnost v aplikaci vyhledávat, která zde momentálně chybí. Uživatel by tak mohl prohledávat své projekty, záležitosti i úkoly, vyhledávat ostatní uživatele a tak dále.

5.2.10 Notifikace

Notifikace v aplikaci slouží jako způsob informování uživatele o důležitých změnách, které v projektech na kterých spolupracuje provedli například jeho spolupracovníci.

Notifikace se zobrazují na vrchním panelu pod ikonkou zvonečku s případným číslem informujícím uživatele o počtu nezobrazených notifikací. Příklad tohoto zobrazení je uveden v příloze na obrázku číslo C.4.

V tomto panelu jsou zobrazeny pouze dosud nepřečtené notifikace a notifikace přečtené před méně než třemi dny. Během budoucího vývoje by bylo vhodné vytvořit samostatnou obrazovku pro všechny notifikace, kde by si uživatel mohl prohlížet i starší notifikace a případně by mohl měnit i nastavení jejich zobrazování.

V současné době obdrží uživatel v aplikaci upozornění při těchto událostech:

- Při rozesílání pravidelných týdenních připomínek (více podkapitola 5.2.8).
- Při smazání projektu nebo záležitosti, které byl uživatel součástí.
- Při archivaci projektu, jehož byl uživatel součástí.
- Při duplikaci projektu, když obdrží práva k novému (zduplikovanému) projektu.
- Pokud byl pozván k novému projektu či záležitosti.

- Pokud byl přiřazen ke konkrétnímu úkolu.
- Pokud mu byla odebrána práva k projektu či záležitosti.
- Pokud jiný uživatel opustil projekt, kterého je administrátorem.
- Pokud se blíží čas konce projektu, k němuž je přiřazen.

5.2.11 Přizpůsobení menším obrazovkám

Při vývoji této aplikace byl kladen velký důraz na možnost zobrazit a využívat všechny její funkce na všech nejčastějších zařízeních a obrazovkách. Každá vytvořená obrazovka tak byla testována na všech obvyklých velikostech obrazovky a případně byla část obsahu upravována tak, aby se na všech velikostech zobrazovala správně.

Pro implementaci této funkcionality bylo využito vlastností knihovny **Bootstrap** a komponentů typu **container** a systému **Grid**. Ten umožňuje rozdělit obrazovku do mřížky o až 12 sloupcích a neomezeném počtu řádků. Tato mřížka je potom systémem sama přeskupována tak, že na velkých obrazovkách, jako je například obrazovka počítače, jsou všechny sloupce v jednom řádku vedle sebe, zatímco na menších obrazovkách, jako jsou například displaye tabletů či mobilů, jsou tyto sloupce postupně skládány pod sebe tak, aby byl jejich obsah vždy dostatečně velký a čitelný.

Zároveň jsem také využíval systému **responsive breakpoints** knihovny **Bootstrap**, který umožňuje měnit vlastnosti některých prvků v závislosti na aktuální šířce obrazovky. Díky tomuto je aplikace schopná na menších obrazovkách například skrývat některé méně důležité prvky, nebo třeba jiné prvky zobrazovat jinak veliké.

Na obrázku v příloze číslo C.10 je vidět příklad zobrazení detailu projektu na menší obrazovce. Té je přizpůsoben vrchní navigační panel i umístění jednotlivých prvků na této stránce: například komentáře jsou místo vedle grafu zobrazeny pod ním a zobrazení termínů je zjednodušeno.

Kapitola 6

Testování

6.1 Způsob testování

Testování aplikace probíhalo dvěma způsoby. V první fázi probíhalo testování s jednotlivými uživateli ve věku od 20 do 50 let. Těmto uživatelům byla aplikace ukázána (často v různé fázi implementace), byli s nimi probírány jejich názory na aplikaci a její funkce. S každým z těchto uživatelů byl zároveň vytvořen společný projekt na kterém jsem se s nimi snažil alespoň po několik dní spolupracovat. Závěry tohoto testování byly pro vývoj aplikace velmi důležité, protože mi pomohly zlepšit některé navržené prvky aplikace a zároveň mi poskytly cenné názory na aplikaci a projektové řízení obecně. Některé z nejdůležitějších získaných nápadů jsou pak popsány v sekci Výsledky testování.

Během druhé fáze testování již byla aplikace plně funkční a proto jsem se rozhodl oslovit tři z možných cílových skupin aplikace a ty jsem požádal o dlouhodobější testování aplikace.

6.1.1 Malá firma

Malá až střední firma, která dlouhodobě hledá vhodný způsob jak spravovat své projekty. Alespoň někteří zaměstnanci již mají zkušenosti i s jinými aplikacemi na řízení projektů a znají jejich funkce. Firma má přes 50 zaměstnanců, ale nepočítá s tím, že by aplikaci používalo více než 20 lidí, natož pak na jednom projektu.

I přesto firma pracuje s velkým množstvím složitých projektů, které se skládají z množství podúkolů. Tyto projekty se navíc velmi často opakují (jednou měsíčně, ročně, ...).

Výsledkem testování v této konkrétní firmě byl závěr, že pro ně mnou navržená aplikace není příliš vhodná. Aplikace umožňuje mít pouze čtyři úkoly u každé záležitosti (z důvodu jejich reprezentace pomocí grafu), ale jejich projekty často měly mnohem složitější strukturu. Také jim chyběla možnost nastavení kritické cesty úkolu: navazování jednotlivých úkolů na sebe a možnost v případě zpoždění odsunout i termíny všech navazujících úkolů (i když tuto funkcionalitu, pokud vím, nenabízí žádná větší známá aplikace pro správu projektů).

6.1.2 Skupina přátel

Malá skupina přátel o maximálně pěti lidech, která pořádá společnou zahradní oslavu. Vedoucí této skupiny s počítači pracuje, ostatní členové však počítač používají maximálně k zábavě a nejsou příliš velcí nadšenci do moderních technologií.

Oslava by se dala zcela jistě uspořádat i bez správy projektu (jako koneckonců každý projekt), ale skupina si od použití této aplikace slibovala spolehlivé zařízení všech úkolů a lepší informovanost ostatních členů o průběhu příprav.

Výsledkem byla maximální spokojenost všech uživatelů. Ačkoliv nastavení aplikace není příliš složité, vytvořil vedoucí všechny úkoly a ostatní členové pouze zaznamenávali svůj pokrok. Tyto změny jim zabraly vždy maximálně půl minuty a mohli je provádět i z mobilních zařízení, například cestou z nákupu. U uživatelů navíc nebylo ani nutné, aby se do aplikace přihlašovali – změny mohli zaznamenávat pod anonymním účtem a se správou uživatelských oprávnění si nedělali velké starosti. Ani počítačově méně zdatní členové neměli problém s pochopením principů aplikace a nedošlo tedy k žádným významným komplikacím.

6.1.3 Skupina studentů

Skupina čtyř studentů studujících Fakultu informačních technologií. Někteří z nich mají již předchozí zkušenosti s jinými aplikacemi pro správu projektů.

Od této aplikace očekávají jednoduchou správu úkolů na společném projektu a možnost kontroly ostatních členů, zda svou práci nezanedbávají.

V závěru testování byli studenti s aplikací vesměs spokojeni. Vytvoření projektu i zaznamenávání úkolů bylo dle jejich slov jednodušší než ve všech ostatních aplikacích, se kterými měli dosavadní zkušenosti. Také kontrola ostatních členů týmu fungovala dle očekávání. Přesto jim v aplikaci chyběla například možnost vytváření a spravování týmů nebo propojení s aplikacemi jako je GitHub nebo Google Drive, které znají z jiných aplikací.

6.2 Výsledky testování

V následujících podkapitolách popisují nejdůležitější objevené nedostatky a připomínky, které se během obou fází testování vyskytovaly nejčastěji.

6.2.1 Přidání termínů

Během počáteční fáze testování aplikace mi nezávisle na sobě několik uživatelů sdělilo, že pro plnohodnotné sledování úkolů potřebují mít možnost v aplikaci k úkolům přidávat termíny jejich dokončení. Jelikož se jedná o téměř základní funkcionalitu, která měla být vytvořena již při návrhu aplikace, je její dodatečný návrh popsán v kapitole 4.3.1.

Původně byly termíny navrženy tím způsobem, že každá záležitost i každý její úkol mohl mít neomezený počet vlastních termínů. Tento návrh počítal s případem, kdy by si uživatel přál vytvořit například samostatný termín pro první polovinu úkolu a druhý termín pro zbytek úkolu. Při testování se ovšem tento přístup nesetkal s pochopením uživatelů a návrh byl nakonec upraven tím způsobem, že každá záležitost a každý úkol může mít maximálně jeden termín. Tento přístup navíc výrazně zjednodušil uživatelské rozhraní pro vytváření a úpravu termínů a s touto verzí implementace již byli všichni uživatelé spokojeni.

Zároveň byly původně termíny seřazeny u každé záležitosti podle data jejich ukončení, tedy nejdříve byly seřazeny termíny s nejbližším datem bez ohledu na to, zda se termín

vztahuje k celé záležitosti, či pouze k jednomu z úkolů u dané záležitosti. Některým uživatelům ale tento způsob nepřišel intuitivní a ve výsledné implementaci je tedy nejdříve zařazen termín celé záležitosti a pak jsou teprve podle data ukončení seřazeny jednotlivé termíny úkolů.

6.2.2 Přehled mých termínů

Přidání termínů popsané v předchozí podkapitole a jejich zobrazení na stránce záležitosti výrazně pomohlo ke zorientování se v časovém plánu dané záležitosti, což bylo užitečné hlavně pro vedoucí pracovníky, kteří na záležitost dohlíželi, ale nebylo nejlepším řešením pro samotné uživatele. Ti si během testování stěžovali, že hlavně při větším množství projektů a záležitostí již zapomínají co vše mají na starosti a na jaký úkol by se měli zaměřit.

Do aplikace tedy byl přidán přehled termínu daného uživatele, jehož návrh je popsán v podkapitole 4.3.1. Tento přehled vybere všechny termíny, které náleží právě přihlášenému uživateli, seskupí je podle záležitosti (sloučí do jedné skupiny termíny všech uživatelových úkolů na dané záležitosti) a seřadí je dle jejich termínu dokončení od nejnaléhavějších až po ty nejvzdálenější.

Nejnáléhavější termíny z tohoto přehledu byly kromě samostatné stránky v aplikaci zobrazeny také přímo na vrcholu stránky s uživatelovou nástěnkou (jejíž implementace je popsána v podkapitole 5.2.2) tak, aby uživatel ihned po přihlášení do aplikace věděl na čem by měl pracovat a kolik času mu zbývá.

6.2.3 Přiřazení uživatele k úkolu

Původní návrh uživatelských práv k záležitosti počítal pouze se třemi možnostmi: uživatel má buď plná administrátorská práva, nebo může přidávat a upravovat všechny úkoly, ale nemůže měnit některá další nastavení (například záležitost mazat), případně si vše může pouze prohlížet, ale nemůže měnit vůbec nic.

Během testování ve firemním prostředí jsem ovšem narazil na případ, kdy vedoucí jednotlivým pracovníkům buď nevěřil, nebo jim prostě nechtěl umožnit měnit jiné úkoly (třeba i na stejné záležitosti), než ke kterým dostal výslovná práva. Tento požadavek ovšem vyžadoval změnu systému uživatelských práv, rozšíření nastavení aplikace i zásah do vykreslování grafu.

Pro jeho provedení bylo za prvé potřeba umožnit administrátorovi přiřazovat uživatele k jednotlivým úkolům. Tato možnost byla nakonec přidána na dvě různé obrazovky současně, protože na obou obrazovkách tato volba dává smysl a bez této duplikace byli uživatelé při testování často zmatení a nevěděli, kde by danou možnost mohli hledat.

Na stránku s možností upravit jednotlivé úkoly v záležitosti (s možností například měnit název nebo barvu úkolu popsané v podkapitole 5.2.7) byl přidán řádek s možností přiřadit k úkolu uživatele. Tato možnost je implementována pomocí textového pole, kde může správce přidat jednak uživatele, který již na projektu spolupracuje, ale také zde může zadat naprosto nového uživatele či pouze jeho email a tento uživatel pak bude k projektu přidán s právy pouze k úpravě jednoho daného úkolu. Druhou stránkou je pak seznam s uživateli, kteří již jsou k úkolu přiřazeni popsaný v podkapitole číslo 5.2.5. Na tento seznam byl ke každému uživateli přidán seznam úkolů vytvořených u dané záležitosti a administrátor tak může každému uživateli jednoduše přiřadit jeho úkoly. Přiřazovat uživatele k úkolům tak lze z obou stran, tedy přiřadit úkol k uživateli i přiřadit uživatele k úkolu.

Pro tuto funkci bylo také nutné upravit graf s jednotlivými úkoly. Původně bylo jméno úkolu zobrazeno pod grafem, ale s touto funkcí bylo přesunuto přímo do grafu nad daný

sloupeček a pod grafem je pak vždy napsáno jméno uživatele, kterému daný úkol náleží. Zároveň bylo potřeba upravit práva k úpravě úkolu, které byly do této doby řešeny velmi jednoduše: měl-li uživatel práva k zápisu, načetla se vlastní knihovna upravující chování grafu po najetí a táhnutí myši a v opačném případě se načetla pouze knihovna pro zobrazení grafu. Po této úpravě je potřeba této knihovně předat i pole s právy k jednotlivým úkolům a po najetí uživatelem nad graf daného úkolu aplikace zkontroluje, zda má práva k jeho úpravě a až poté začne případně překreslovat graf dle aktuální polohy kurzoru v grafu.

6.2.4 Duplikace projektu

Další funkcionalitou často požadovanou při testování bylo vytváření šablon projektů. Tato vlastnost je výhodná hlavně v případě, pokud by např. firma chtěla sledovat často se opakující projekty. Může se jednat například o případ, kdy zadává do katalogu nový produkt. Tento postup je často velmi podobný a často se na něm podílejí i stejní zaměstnanci (produkt je potřeba nafotit, vytvořit grafiku do katalogu, napsat a přeložit texty na webové stránky, ...), ale vytváření složité a stále se opakující struktury projektu je zdoluhavé a zbytečné.

Návrh této funkce včetně jejích chování je popsán v podkapitole 4.3.2.

Během testování si uživatelé tuto funkci velmi pochvalovali, jelikož jim za prvé ušetřila čas při zadávání projektu, ale hlavně ji prý nenašli ve vyhovující formě v žádné konkurenční aplikaci na sledování úkolů.

6.2.5 Návaznost úkolů

Mezi požadavky na pokročilejší funkcionalitu, které vyplynuly z testování aplikace, patří žádost o možnost řetězit jednotlivé úkoly. Jedná se o situaci, kdy jeden úkol je přímo závislý na dokončení jiného úkolu a nemůže začít, pokud nebyl předchozí úkol dokončen (jako příklad může opět posloužit například zpracování grafiky pokud grafik nemá nafocené materiály).

Pro implementaci tohoto požadavku by bylo potřeba vytvořit možnost v nastavení, kde by administrátor mohl přidat prerekvizity libovolnému úkolu (a to i z jiných záležitostí stejného projektu) a bylo by třeba přidat grafický prvek, který by signalizoval, že úkol ještě čeká na splnění jiného úkolu a nemůže tedy být započat. Otázkou je, zda by aplikace měla být striktní a vůbec nepovolit jeho změnu, pokud nejsou všechny jeho prerekvizity na hodnotě 100%, nebo zda by se mělo jednat pouze o „kosmetický“ prvek.

Tento požadavek nebyl v současné době v aplikaci implementován, ale může se jednat o jeden ze způsobů, jakým lze aplikaci dále rozvíjet. Nicméně implementace této funkce nepřináší aplikaci žádnou velkou funkcionalitu navíc a neměla by dle mého názoru v budoucnu mít nejvyšší prioritu.

6.2.6 Odložení úkolu

K požadavkům na návaznost úkolů a na přidání termínů připadá také požadavek na odložení termínu úkolu. Jedná se o případ, kdy vedoucí stanoví termín nějakého úkolu a jeho vykonavatel jej nestihne vyplnit včas. Ten by pak měl mít možnost do aplikace zadat, že se daný termín nezvládl splnit. Tento krok by pak měl vyvolat několik akcí: vykonavatel by měl mít možnost popsat důvod nesplnění úkolu v daném termínu a zadat čas prodloužení termínu, jeho vedoucí by měl být o této události náležitě informován a všechny úkoly, které

na tento úkol přímo i nepřímo navazují (viz předchozí podkapitola Návaznost úkolů) by měly být automaticky o zadaný čas také prodlouženy (a jejich majitelé informováni).

Tento požadavek zatím, stejně jako předchozí požadavek, v aplikaci nebyl implementován a jedná se tak o návrh na možné rozšíření aplikace do budoucna.

6.2.7 Integrace s dalšími aplikacemi

Další vlastností, která při testování aplikace chyběla většímu množství uživatelů je integrace s cizími aplikacemi. Může se například jednat o automatické zadávání termínu do Google kalendáře, sdílení souborů z Dropboxu či Google Drivu, nebo automatické odesílání nových příspěvků na aplikace jako je třeba Discord.

Tato funkcionality uživatelům chyběla o to více, že většina konkurenčních komerčních aplikací tuto integraci nabízí. Například Trello nabízí integraci s Evernote, Githubem, Google Hangouts, Mailchimpem, Salesforce, Slackem, Google Diskem, Dropboxem a dalšími. . . (to vše ovšem pouze v placené verzi za 10 dolarů měsíčně), Asana nabízí integraci s více než 60 různými aplikacemi atd.

Konkurovat v tomto ohledu takto velkým aplikacím s takto širokou nabídkou je v rámci této diplomové práce téměř nemožné, bohužel to však může být jeden z důvodů, proč by si nakonec uživatelé mohli zvolit některou z konkurenčních aplikací. Při budoucím vývoji by ovšem mělo být prioritou vybrat alespoň několik nejužitečnějších a nejpoužívanějších aplikací a integrovat je s touto prací tak, aby se uživatelům jejich možnost pohodlně sledovat postup na svých záležitostech co možná nejvíc zjednodušila.

Kapitola 7

Závěr

V úvodu této práce byly nastudovány obecné principy tvorby produktů kladoucích důraz na snadnou uživatelskou použitelnost a způsoby aplikování těchto principů při vytváření webových aplikací.

Dále tato práce studovala problematiku *project managementu* a monitorování splnění úkolů definovaných uživatelem. Byly nastudovány nejčastěji používané webové aplikace zabývající se touto problematikou. U každé z těchto aplikací byly zhodnoceny její klady a zápory. Zároveň jsem se pokusil identifikovat jejich silné i slabé stránky, abych je mohl následně zohlednit při návrhu aplikace.

Na základě zadání jsem navrhl aplikaci, která se částečně vymyká nejznámějším aplikacím tím, že je orientovaná na proces vývoje projektu a umožňuje plynule měnit vnímaný pokrok na úkolu od jeho začátku až do konce. Navržená aplikace umožňuje na první pohled vidět stav jednotlivých úkolů a poskytuje tak jiný způsob pohledu na řízení projektů. Takto navržená struktura také dovoluje uživateli měnit hodnotu úkolů tak, aby mohla i klesat, čímž lze mimo jiné zaznamenávat i subjektivní hodnoty typu „jak mě práce těšila“.

Aplikace byla implementována s cílem být maximálně jednoduchá na používání a přitom tak, aby stále umožňovala plnohodnotnou správu projektů. Aplikace nabízí jednoduchý způsob sdílení projektů i nastavení uživatelských práv a je v ní také možné začít spolupracovat okamžitě i bez nutnosti registrace, vyžaduje-li to projekt. Aplikace je vysoce interaktivní a dbá na jasnou vizuální reprezentaci stavu jednotlivých úkolů, přičemž úprava postupu na daném úkolu nezabírá více než pár sekund. Jsou v ní také implementovány další funkce jako je nápověda pro nové uživatele.

Výsledná aplikace byla nasazena na webový server a testována na několika různých skupinách uživatelů. Mezi hlavní výhody aplikace patří její jednoduchost a přívětivost vůči novým uživatelům. Tato práce však nemá ambice vyrovnat se velkým aplikacím s velkým množstvím funkcí, které budou vždy pro některé uživatele lákavější a vhodnější.

Dalším krokem ve vývoji by mělo být nasazení projektu na vlastní doménu, následná aktivní propagace a nabídnutí velkému množství uživatelů. Potenciálním směrem, kterým by se aplikace mohla dále rozvíjet je také například integrace s externími aplikacemi typu Google Calendar, Dropbox a další, či snadná správa celých týmů.

Průzkumem trhu jsem zjistil, že na jednu stranu existuje solidní povědomí o potřebě řešení mnoha situací (profesních i soukromých) pomocí projektového řízení a na druhou stranu neochotu k učení se i soběméně složitých aplikací a právě v jednoduchosti vidím největší výhodu mé aplikace.

Literatura

- [1] Chart.js. [Online; cit. 16.1.2018].
URL <http://www.chartjs.org/>
- [2] The Doctrine Project. [Online; cit. 16.1.2018].
URL <http://www.doctrine-project.org/>
- [3] Podio. [Online; cit. 16.1.2018].
URL <https://podio.com/>
- [4] Symfony Project. [Online; cit. 16.1.2018].
URL <http://symfony.com/>
- [5] Trello. [Online; cit. 16.1.2018].
URL <https://trello.com>
- [6] Forrester Consulting; Akamai Technologies: eCommerce web site performance and its correlation with an online shopper's behavior. September 14, 2009.
- [7] Huff Industrial Marketing, KoMarketing, & BuyerZone: 2015 B2B Web Usability Report: What B2B Buyers Want from Vendor Websites. April 2015.
- [8] Colborne, G.: *Simple and Usable Web, Mobile, and Interaction Design*. Thousand Oaks, CA, USA: New Riders Publishing, první vydání, 2010, ISBN 0321703545, 9780321703545.
- [9] ComScore: The 2016 U.S. Cross-Platform Future in Focus. 30. 3. 2016, [cit. 14.1.2018].
- [10] Ibbs, C. W.; Kwak, Y.-H.: Assessing project management maturity. *Project Management Journal*, ročník 31, č. 1, 2000: s. 32–43.
- [11] ISO 6385:2016: Ergonomics principles in the design of work systems. Standard, International Organization for Standardization, 2016-09.
- [12] ISO 9241-11:1998: Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability. Standard, International Organization for Standardization, 1998-03.
- [13] ISO 9241-210:2010: Ergonomics of human-system interaction – Part 210: Human-centred design for interactive systems. Standard, International Organization for Standardization, 2010-03.
- [14] Janíková, S.: *Nástroj pro sledování postupu na záležitostech*. 2017.

- [15] Kerzner, H.: *Project Management*. Hoboken, New Jersey: John Wiley & Sons, Inc., 11 vydání, 2013, ISBN 978-1-118-02227-6.
- [16] Moskovitz, D.; Rosenstein, J.: Asana. [Online; cit. 16.1.2018].
URL <https://asana.com>
- [17] Norman, D. A.: *The Design of Everyday Things*. New York, NY, USA: Basic Books, Inc., 2002, ISBN 9780465067107.
- [18] Norman, D. A.; Draper, S. W.: *User Centered System Design; New Perspectives on Human-Computer Interaction*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 1986, ISBN 0898597811.
- [19] Otto, M.: Bootstrap. [Online; cit. 16.1.2018].
URL <https://getbootstrap.com/>
- [20] Ritter, F. E.; Baxter, G. D.; Churchill, E. F.: *Foundations for Designing User-Centered Systems: What System Designers Need to Know About People*. Springer Publishing Company, Incorporated, 2014, ISBN 144715133X, 9781447151333.
- [21] Schenker, M.: 7 Simple Ways to Achieve a User-Centered Website. Červenec 2014, [Online; cit. 28.12.2017].
URL <https://www.webdesignerdepot.com/2014/07/7-simple-ways-to-achieve-a-user-centered-website/>
- [22] Travis, D.: ISO 13407 is dead. Long live ISO 9241-210! Červen 2011, [Online; cit. 12.01.2018].
URL <https://www.userfocus.co.uk/articles/iso-13407-is-dead.html>
- [23] U.S. Department of Health & Human Services: User-Centered Design Basics. [Online; cit. 14.01.2018].
URL <https://www.usability.gov/what-and-why/user-centered-design.html>
- [24] Řezáč, J.: *Web ostrý jako břitva*. Baroque partners, s.r.o., 2014, ISBN 978-80-87923-01-6.

Příloha A

Obsah přiloženého paměťového média

Přiložené CD obsahuje:

- Technickou zprávu ve formátu PDF
- Zdrojové kódy technické zprávy ve složce `/zprava`
- Zdrojové kódy aplikace ve složce `/aplikace`

Příloha B

Odkazy na aplikaci

Odkazy na webovou stránku s aplikací a na repozitáře se zdrojovými kódy.

Webová adresa aplikace:

- <https://xlamac00-dip.herokuapp.com/>

Zdrojové kódy aplikace:

- <https://github.com/Xlamac00/DIP2>

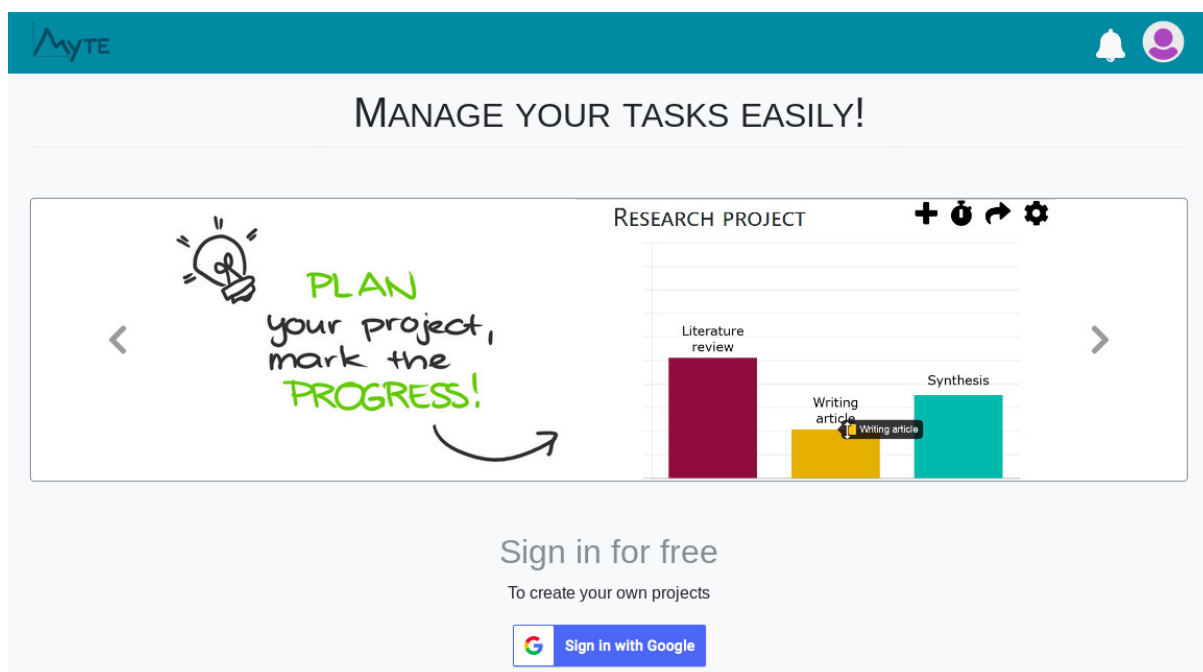
Zdrojové kódy technické zprávy:

- <https://www.overleaf.com/read/rbhdmtkymqfv>

Příloha C

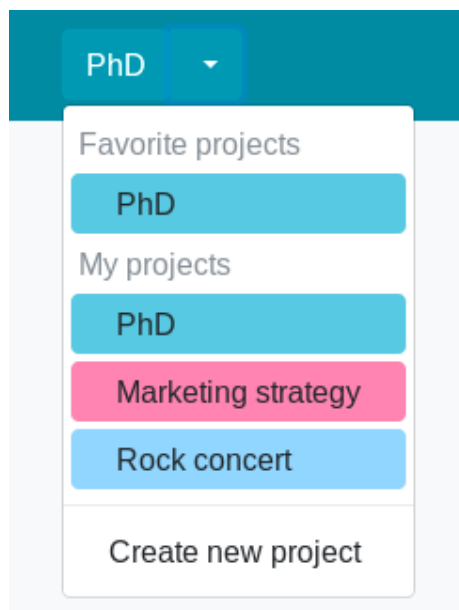
Další prvky uživatelského rozhraní

C.1 Titulní strana

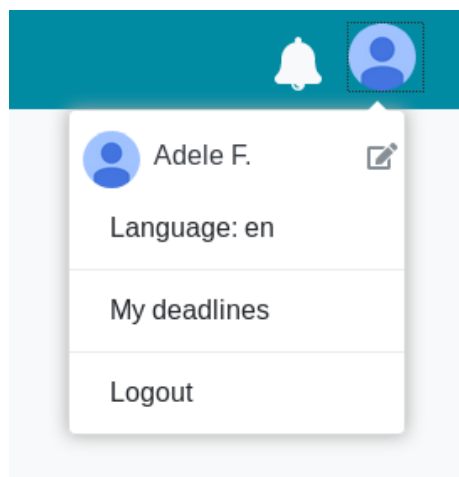


Obrázek C.1: Titulní stránka aplikace pro uživatele bez vytvořených projektů. Jejím cílem je pomocí několika obrázků a animovaných videí představit základní funkce této aplikace a umožnit jim přihlásit se aplikaci používat.

C.2 Navigační panel

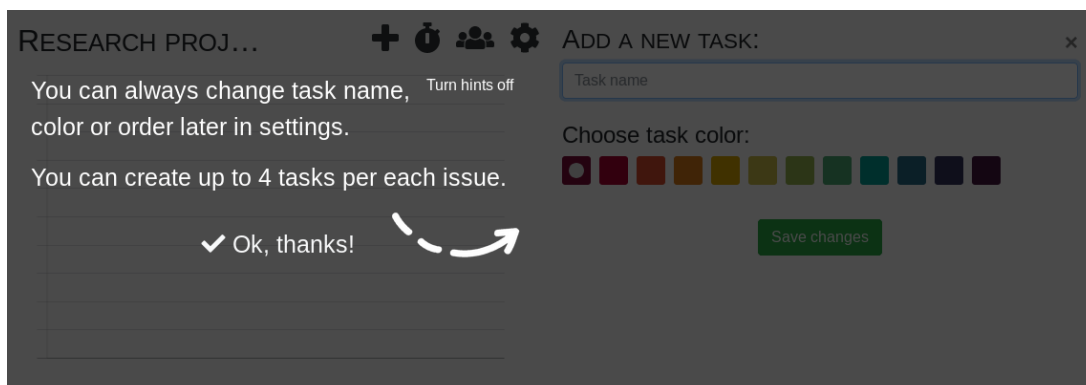


Obrázek C.2: Nabídka, která se zobrazí po kliknutí na ikonku vedle názvu projektu na navigačním panelu. Tato nabídka obsahuje možnost rychle přepínat mezi jednotlivými projekty uživatele a také rychle vytvořit nový projekt.

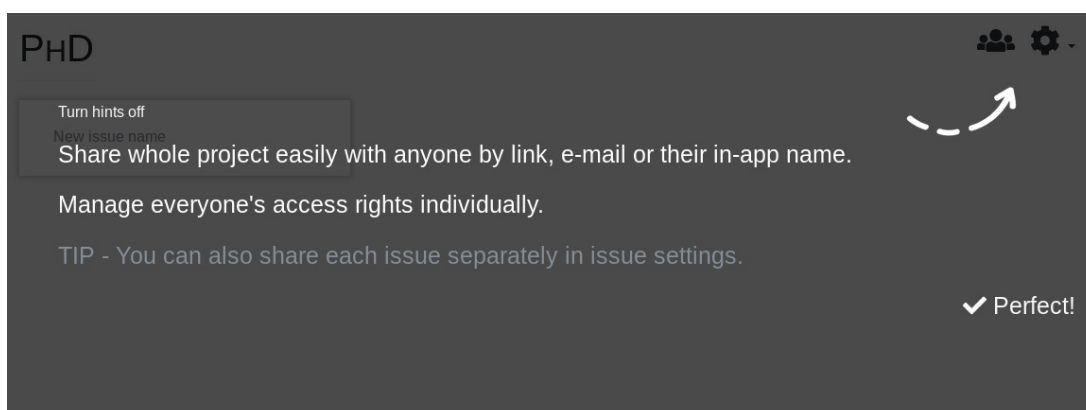


Obrázek C.3: Nabídka, která se zobrazí po kliknutí na uživatelský portrét na navigačním panelu. Nabízí možnost změnit uživatelské jméno a jazyk aplikace. Obsahuje také rychlý odkaz na uživatelské termíny. Anonymním uživatelům nabízí možnost přihlásit se pomocí služby Google, přihlášeným uživatelům pak nabízí možnost odhlásit se.

C.3 Náповěda

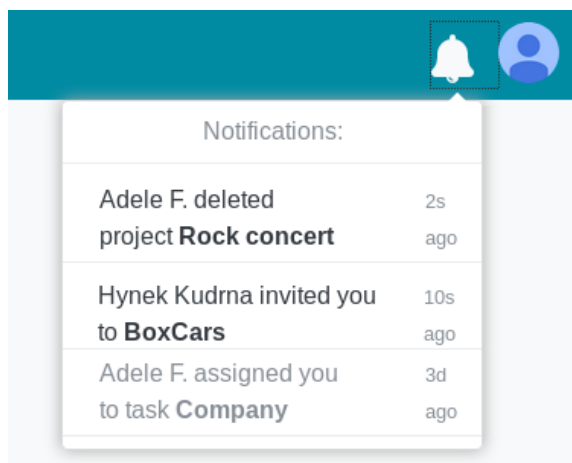


Obrázek C.4: Příklad nápovědy zobrazené uživateli při tvorbě nové záležitosti. V této nápovědě je uživatel vyzván k vytvoření alespoň jednoho úkolu, je mu vysvětleno, že úkoly může později upravovat v nastavení a že aplikace umožňuje vytvořit pouze 4 úkoly u každé záležitosti.



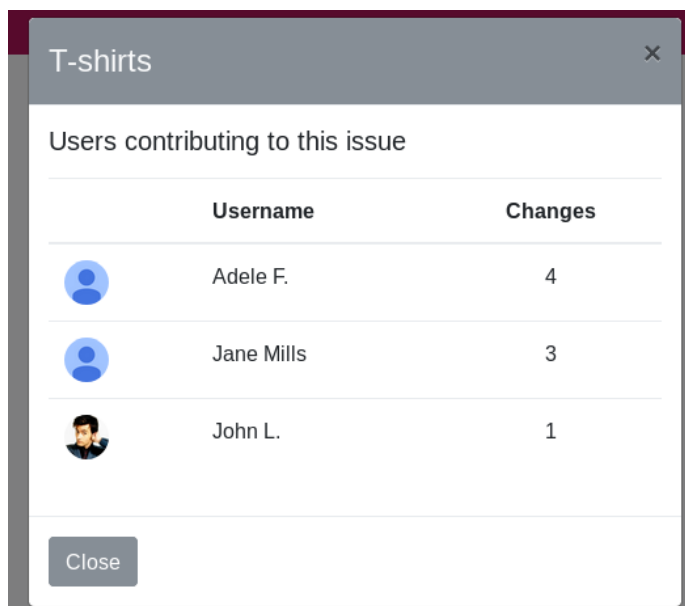
Obrázek C.5: Příklad další nápovědy, která tentokrát vysvětluje uživateli způsob sdílení projektů. Každá nápověda se snaží vysvětlit jeden z prvků aplikace pomocí co nejjednodušších vět.

C.4 Notifikace



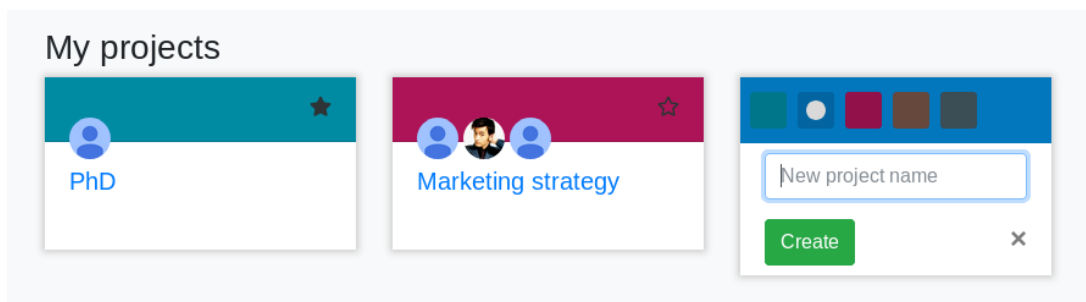
Obrázek C.6: Příklad zobrazení doručených notifikací v aplikaci. Tmavou barvou jsou zobrazeny nově přečtené notifikace, šedou jsou poté zobrazeny všechny notifikace přečtené před méně než třemi dny.

C.5 Seznam uživatelů



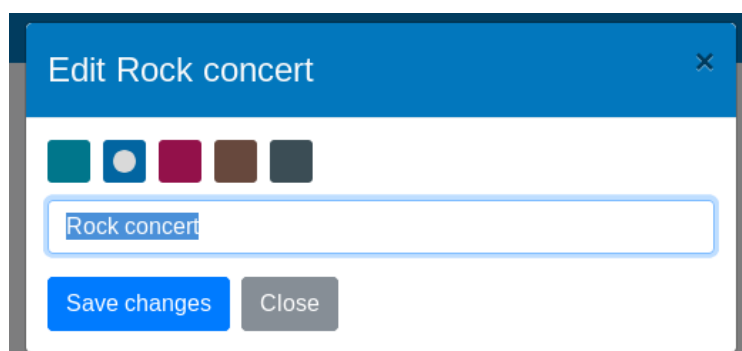
Obrázek C.7: Seznam všech uživatelů přispívajících k vybrané záležitosti seřazených podle počtu jejich příspěvků. Toto modální okno se zobrazí po kliknutí na seznam uživatelů u karet projektu nebo záležitosti. Toto okno je pro projekt i záležitost téměř shodné.

C.6 Karta k založení nového projektu



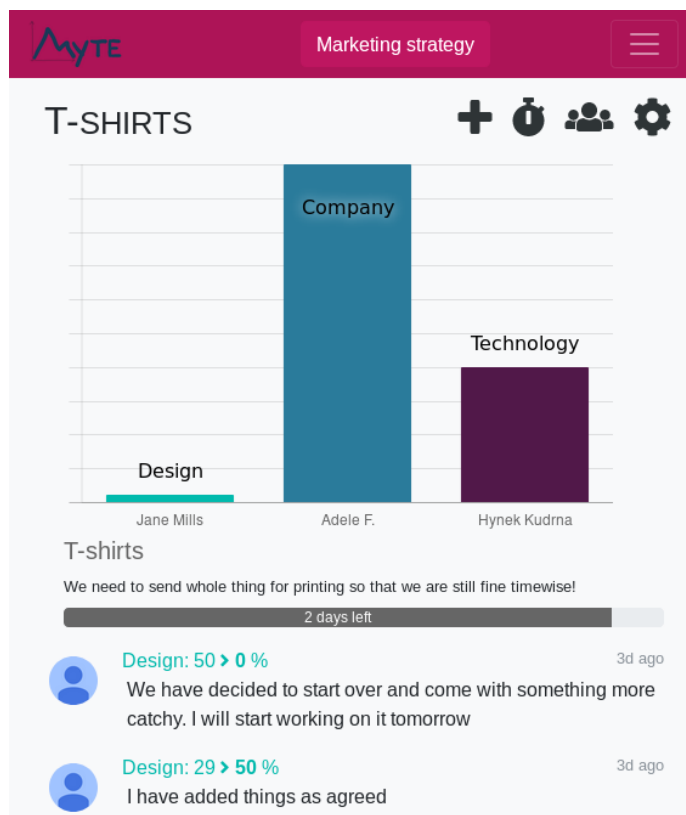
Obrázek C.8: Nástěnka uživatele obsahující seznam karet všech uživatelských projektů s poslední kartou umožňující vytvoření nového projektu. Po kliknutí na tuto kartu se uživateli zvýrazní pole pro zadání nového jména projektu a zároveň uživatel dostane možnost změnit barvu projektu tak, jak je zobrazeno na tomto obrázku.

C.7 Editace projektu



Obrázek C.9: Modální okno pro editaci projektu umožňující změnit barvu a název projektu. Stejný dialog je využit i pokud uživatel klikne na tlačítko „create new project“ umístěné v rozbalovacím menu ve vrchní navigační liště.

C.8 Přizpůsobení menším obrazovkám



Obrázek C.10: Příklad zobrazení stránky s detailem záležitosti na obrazovce s menší šířkou displaye.